ORTHOGONALIZATION STRATEGY FOR FASTER RECONSTRUCTION FROM PROJECTIONS

A Thesis
Submitted In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By

LAV GUPTA

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

CERTIFICATE

This is to certify that the thesis entitled "Orthogonalization Strategy for Faster Reconstruction from Projections" is a record of the work carried out under my supervision and that it has not been submitted elsewhere for a degree.

(S.K. Mullick)
Professor
Department of Electrical Engineering
Indian Institute of Technology
Kanpur.

ACKNOWLEDGEMENT

I am highly thankful to my thesis supervisor, for suggesting an interesting topic and constantly inspiring, encouraging and guiding me throughout the course of this work.

Thanks are also due to my friends Rashmi & Rajiv for doing all the proof reading and to Akhil for helping me in the preparation of the final phase of my thesis. I should not forget to mention Girish & 'Opu' who provided constant moral support.

I am grateful to Neeta who kept on coaxing me to finish the thesis fast.

Finally I thank Mr. C.M. Abraham for his neat & efficient typing.

Lav Gupta

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Projection algorithm has been found to have a number of desirable properties which make the algorithm attractive for practical applications. In this work the algorithm has been reviewed in the context of reconstruction problem. Effects of binary orthogonalization on the speed of convergence of this algorithm have been numerically studied using a set of 32x32 digital functions. It is shown that orthogonalization does expedite convergence of the projection method and results in a securing of about 30-50% in computational time.

CHAPTER 1

INTRODUCTION

The 1979 Nobel Prize for Medicine and Physiology was shared by Allan M Cormack and G N Hounsfield for the development of Computer Assisted Tomography (CAT), a new method for X ray diagnosis. This non invasive diagnostic procedure makes use of computers in imaging internal organs within human body by mathematically combining X-ray image data from numerous angles to obtain the image of any cross-section within the body. The technique involved is 'the reconstruction from projections'. This technique has been applied, in various forms, to other fields of scientific endeavour such as radioastronomy, microscopy, biomedical engineering, cellular biology, nuclear science, digital image transmission etc. The list is limited only by imagination.

In most of the cases images are used to represent the distribution of some property of an object or physical system e.g. molecular structure, X ray. Most images are formed directly by optical instruments using visible light reflected or transmitted by an object. In many applications in which an image is required, only indirect measurements by probing an object can be made with visible radiation or by interpreting radiations emitted by it.

Often the measurement data is not in a suitable form for immediate interpretation. The general aim of all image reconstruction procedures is to process the data to form an image so as to facilitate the interpretation of the measurement.

A number of techniques for reconstruction from projections are available and a brief survey is given in the next section.

1.1 A Word on Present Reconstruction Techniques

J. Radon did the pioneering work in the field of reconstruction and proved that it is possible to reconstruct structures from their projections. The algorithms which have come up in the last few years can be classified into the following :

1. Direct matrix inversion techniques [3] [42]
2. Summation, linear superposition, back projection [4] [43]
3. Algebraic reconstruction techniques (ART) [5] [44]
4. Simultaneous iterative reconstruction technique (SIRT) [6]
5. Orthogonal tangent correction [7]
6. Iterative least squares techniques [3] [8]
7. Convolution techniques [3] [9] [45]
8. Geometric mean iterative techniques [10]
9. Fourier reconstruction [11] [46]
10. Summation of projections after Hilbert transform of the derivative of the projections. [22] [48] [49] [50]

The techniques mentioned above can be grouped broadly into the following four categories :

1. Summation (1)
2. Use of Fourier transforms (7,9)
3. Analytic solution of integral equation (7,9,10)
4. Series expansion approaches (3,4,5,6)

(The numbers in small brackets refer to the serial number of the technique given above).

More generally, these techniques may be grouped as : signal space reconstruction and Fourier space reconstruction.

Bracewell and Riddle [14] proposed algorithm for reconstruction in signal space while De Rosier and A. Klug [11] did the initial work in the area of Fourier transform methods. The Direct Matrix Inversion techniques were dealt with by Budinger and Gullberg [3] [13] who also showed that this method requires large storage and long computational times. Andrews discussed summation, linear superposition, back projection [4] method. Then Vainshtein [16] Ramachandran and Lakshminarayan [20] and Bracewell and Riddle [9] came forward with the convolution method.

Gordon, Bender and Herman [5] and Gordon [21] proposed Algebraic Reconstruction Technique which they claimed to be the best (under certain circumstances) among the contemporary techniques. It was implemented in signal space. Gilbert [6]

came forward with iterative relaxation methods. Krishnan, Prabhu and Krishnamurthy [15] gave an iterative scheme based on unbiased probabilistic estimates. Geometric mean iterative technique was given by P. Schmidlin [10]. D.E. Kuhl proposed orthogonal tangent correction method. In 1971 Vainstein came up with a summation method. Budinger and Goullberg [13] implemented simultaneous reconstruction technique (SIRT). Wernecke and D'Addario [17] have applied the principles of maximum entropy to the two dimensional digital image reconstruction problem.

1.1.1 Choice of Algorithm for our Study

As we have seen above a large number of algorithms exist for reconstructing three dimensional or two dimensional images from their projections. We needed for our study an algorithm which is comparatively well behaved, i.e. its computationally less cumbersome, essentially performs pseudoinversion and is guaranteed to converge. Not much has been said about the relative merits and demerits of various algorithms and we base our decision on whatever comparison is available.

Bellman, Bender, Gordon and Rowe [37] have shown that ART does comparatively better than the Fourier techniques particularly when a limited number of projections in a limited range of angles, is available. Herman [38] compared an improved summation method with ART for 4 test patterns,

came forward with iterative relaxation methods. Krishnan, Prabhu and Krishnamurthy [15] gave an iterative scheme based on unbiased probabilistic estimates. Geometric mean iterative technique was given by P. Schmidlin [10]. D.E. Kuhl proposed orthogonal tangent correction method. In 1971 Vainstein came up with a summation method. Budinger and Goullberg [13] implemented simultaneous reconstruction technique (SIRT). Wernecke and D'Addario [17] have applied the principles of maximum entropy to the two dimensional digital image reconstruction problem.

1.1.1 Choice of Algorithm for our Study

As we have seen above a large number of algorithms exist for reconstructing three dimensional or two dimensional images from their projections. We needed for our study an algorithm which is comparatively well behaved, i.e. its computationally less cumbersome, essentially performs pseudoinversion and is guaranteed to converge. Not much has been said about the relative merits and demerits of various algorithms and we base our decision on whatever comparison is available.

Bellman, Bender, Gordon and Rowe [37] have shown that ART does comparatively better than the Fourier techniques particularly when a limited number of projections in a limited range of angles, is available. Herman [38] compared an improved summation method with ART for 4 test patterns,

4 distinct sets of projections and 18 different criteria of success. ART was consistently better. Herman and Rowland [39] compared ART with SIRT and convolution method. It was found that for underdetermined, noisy system for which projections are available from a narrow range of angles, ART is superior. Schmidlin [40] found ART to be superior to the summation method. Gordon compared convolution and a form of ART and the latter was found to be far superior [41].

The Projection Algorithm (PRA) which was first proposed by Kacmarz [30] as a method of solving linear equations and later reviewed by Tanabe [31] who proved its guaranteed convergence is computationally, in principle, the forerunner of ART. Recently it was applied to the problem of restoration by Huang and Ramakrishna [33] and was found to be very well behaved.

Keeping the above considerations in mind it was felt that PRA could profitably be utilised for the reconstruction problem. Accordingly, this thesis is concerned with computational study of the PRA for the problem in hand.

1.2 Present Work

The reconstruction algorithm which we have mentioned above reconstructs images from their projections with varying speed and accuracy. Some of these algorithms are slow and are suitable only for batch mode processing while

others are faster and allow real time processing. Even at present most of the algorithms are either quite slow or else they require prohibitively large storage. Many practical applications require faster algorithms for the result to be useful.

Ramakrishna [33] advocated the use of some form of orthogonalization with the projection method to expedite its convergence. In the context of the restoration problem it was shown computationally that binary orthogonalization followed by PRA resulted in considerable savings in computational time. Similar techniques have been studied, in this thesis, in the context of reconstruction problem. PRA with orthogonalization, or simply OPRA, has been compared with the PRA to show formers superiority in many respects. The bases of comparison are the number of mismatches, number of projections, number of iterations and some other performance criteria. It has been shown that the gain in computational speed are in the range of 20% - 50%.

1.3 Overview of the Thesis

Chapters 2 and 3 deal with the basic ideas of reconstruction problem in two dimensions. They discuss various steps involved in solving the reconstruction problem digitally. These include digitization, obtaining projections and the formulation of the problem as a set of a large

number of simultaneous linear equations. Chapter 3, in particular, presents the problem in the form in which it will be taken for solution in the later chapters.

Chapter 4 describes the projection method and its speed up by an orthogonalization scheme. Binary orthogonalization has been discussed in some detail.

Chapter 5 contains the results obtained for the test pictures used. A comparison between PRA and OPRA has been made to demonstrate the superiority of OPRA over PRA.

Appendix A discusses reduction of dimensionality of the reconstruction problem, reduction of the storage requirement and of takes up the question of compactness of figures in digital pictures.

Appendix B includes a discussion on the number of views required for reconstruction. It also deals with the limits on discretization of projection angles.

Appendix C gives the listing of the combination of characters used to obtain grey levels in time printer imagery The list of the Fortran programs used in computations is contained in Appendix D.

CHAPTER 2

ON RECONSTRUCTION FROM PROJECTIONS

2.1 Introduction

J. Radon, an austrian mathematician, triggered the development in the field of reconstruction by his historical assertion made in 1917 that a two or three dimensional structure could be reconstructed uniquely from the infinite set of all its projections. Since then, this result has been exploited in developing algorithms for reconstructing images from its projections. Many of the algorithms are mechanisms of evaluating Radon's relation between the value of each picture element in polar coordinates $A(r,\theta)$ and the projections for all angles $P(x,\theta)$, Fig. (2.5), where x denotes an element along the projection corresponding to the line integral through the section to be reconstructed. Thus Radon [22] in 1917 and subsequently others [12] , [49] showed that

$$A(r,\theta) = \frac{1}{2\pi^2} \int_{-\pi/2}^{\pi/2} \int_{-\infty}^{\infty} \frac{p(x,\theta')}{x} \quad \frac{1}{r\sin(\theta-\theta')-x}\, dx\, d\theta' \qquad (2.1)$$

That this method is practically non-implementable can be seen by the requirement of projections at 'all' angles, i.e. infinite number of projections. In practical situations there are only a finite number of views and each measurement is subject to errors. Over the last twenty years specialized techniques have

been developed for solving the problem of estimating the distribution of some property in three dimensional space from many views or projections at various discrete angles. But not many techniques allow an effective compromise between computational time and storage requirements.

2.2 The Versatile Tool

Computer Assisted Tomography (CAT), which won Drs. Allan M Cormack and GN Hounsfield their Nobel prize, involves reconstruction from projections. The technique of reconstruction from projections has been applied, in various forms to other fields such as computer science, theoretical molecular and cellular biology, medical and biological engineering, radiology, optics, crystallography, physics, nuclear science, material science, electrical engineering to name a few.

Algorithms which reconstruct three dimensional object from two dimensional slices are of great importance in biology for they allow us to discover the structure of macromolecules and macromolecular assemblies such as ribosomes [23,24] . In astronomy we can reconstruct the 2-D images of the galaxies from radio signals reaching the earth [14]. In the field of medicine and physiology this method finds a very important application, viz.,detection of tumors and similar growths in human body. Projections are taken at various angles and an image constructed from these brings out the tumor as an area of different (greater)

density as compared to the surrounding tissue. A three dimensional reconstruct can give the size of the tumor and its location with amazing accuracy. Patient studies for isotope distribution in the head, heart and liver can be accomplished by rotating the patient before a scintillation camera in 10° increments. The study time is approximately 30 minutes and the doses are no greater than the routine studies with 100 to 400 mrads. Application of these techniques to the heart involves gating the camera or the computer to overcome motion [13].

Another important application is in the area of digital image transmission. Application of the concept of projections results in a considerable saving in both hardware and processing time.

2.3 Basic Steps in Reconstruction from Projections

We will study here the basic theory behind all techniques of reconstruction using projections before going on to a particular method in later chapters. In the context of this general discussion it is sufficient to define a picture as a real valued, non negative function of two variables. The value of this function at any point gives the grey level at that point [28]. We are considering pictures with grey levels only.

A projection has been defined as an ordered set of values, each being the integral or sum of grey levels along one of a set of parallel rays which are bands of finite width drawn across the

picture (Fig. 2.1). The set $\{ b_1, b_2, \ldots b_j \}$ of the ray sums forms a projection. We will talk about projections and method of obtaining them in the next chapter.

The problem is to reconstruct from a finite subset of its projections taken at distinct angles, the picture under test. The amount of data available is usually less than that required for an exact solution and the use of iterative techniques is preferable [26] if digital processing is to be employed.

2.3.1 Discretization

For digital reconstruction we must discretize the problem. In other words we must sample our picture. One way to sample it is to break it down into small square cells of equal dimensions with the help of a grid of appropriate size. This method is discussed in greater detail later. Another way is to sample the picture with an array of equally spaced points (Fig. 2.2). Any method of sampling will yield some discrete values which can be expressed in the form of an array.

The difficulties faced in practical digitization will be discussed in the next chapter.

With the help of Fig. 2.2 we can understand how, after digitization, the samples are obtained and how the projections are taken. The figure shows an irregular outline which is the two dimensional cross-section of our three dimensional density distribution or activity. The greater the density at any point,

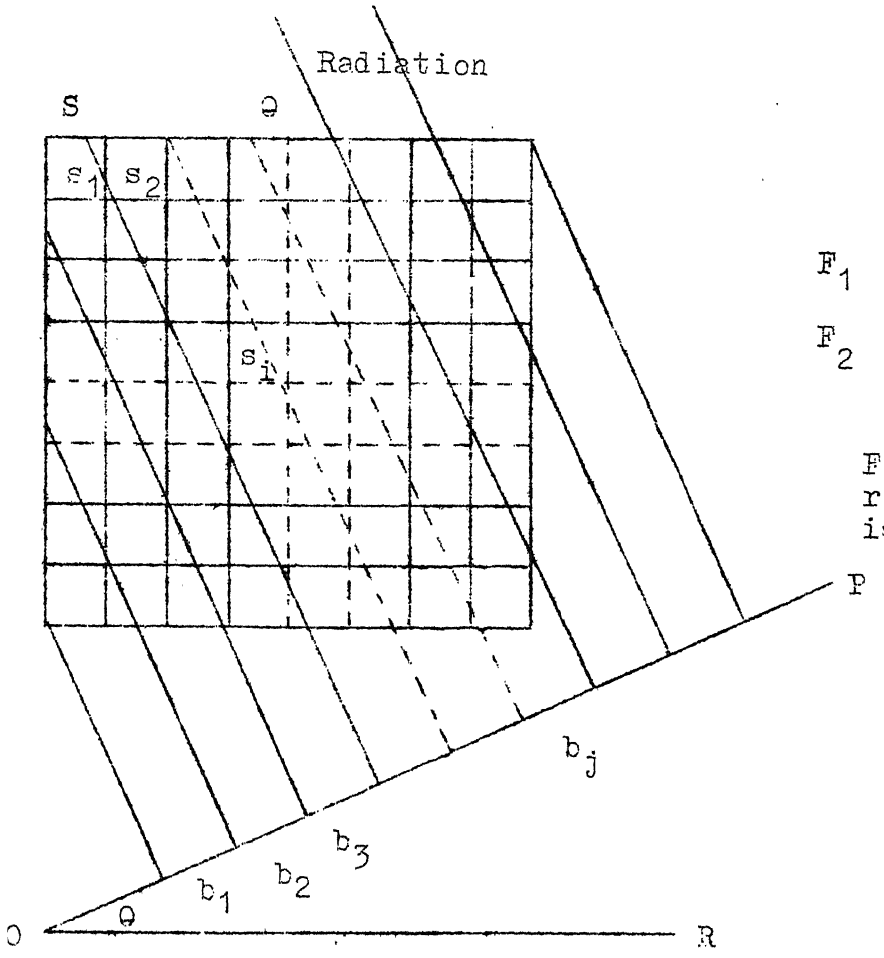


Fig. 2.1 Principles of projection method

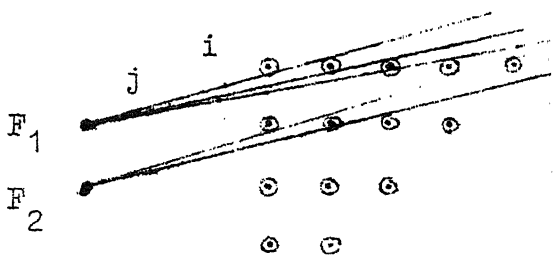


Fig. 2.3 Projections required when point grid is used

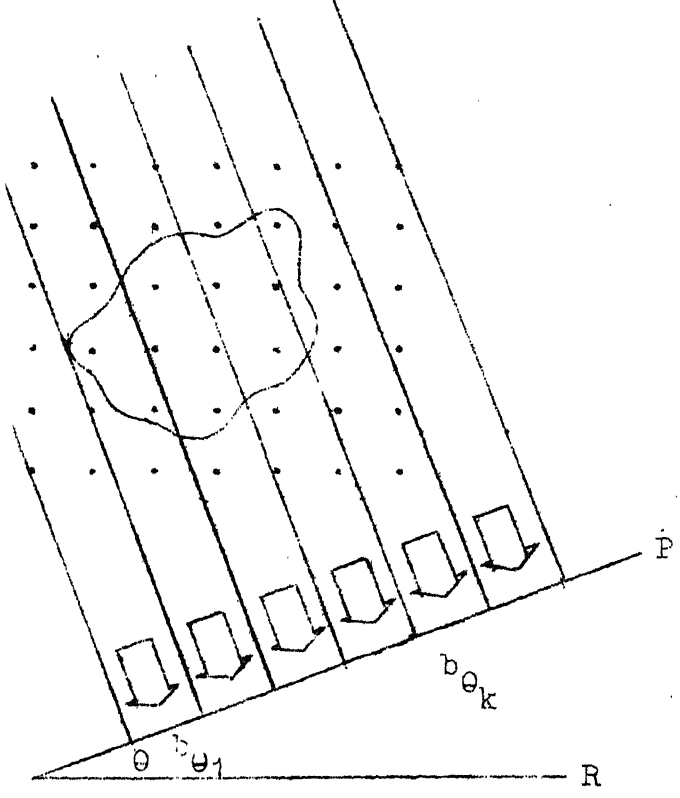


Fig. 2.2 Discretizing and obtaining

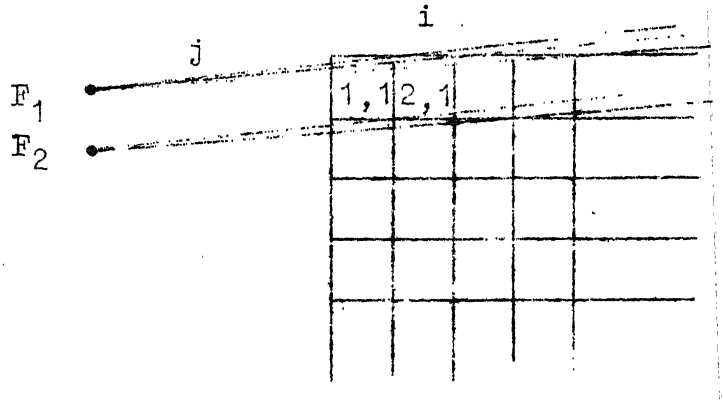


Fig. 2.4 Projections required when square cell grid is used

greater is the activity at that point. The figure also shows, on the plane of cross-section, a nxn array of sampling points which are used to discretize this analog density distribution. The density distribution of the object determines the activity to be assigned to each of the n^2 points of the sampling array. The point falling in the region of greater density gets higher activity attached to it and so on. We will see in the next section how we use this discretized distribution for obtaining projections. The terms density and activity will be used synonymously in the text.

2.3.2 Obtaining Projections

To get projections, the activity in the plane of cross-section i.e. the activity associated with all the points, is projected onto a line (OP in Fig. 2.2) at angle θ_k to some reference line (OR in Fig. 2.2). One can see in the same figure a set of parallel rays δ_1, δ_2, δ_3 ... which are all parallel to each other and to the plane of cross-section. Not much observation is required to make out that any of the sampling point can fall, in general, in only one of these rays. If it falls on the edge then it is associated with the ray to which it contributes most. The sum of activities of all the sample points in any ray is known as the ray sum of that ray. To be more specific we can think of a projection as being formed by a group of parallel rays. The projection value associated with

ray kΘ is $b_{\Theta k}$ which is the sum of the activities of all the points included in the ray. The set of these values form a projection. Other projections are got by projecting the activities a_{ij} onto the line OP at different angles to the same reference line OR. The set $b_{\Theta i}$, i = 1, ... n forms a projection at an angle Θ.

The problem now boils down to reconstruction of the true values of the activities a_{ij} (i:1,n, j:1,n) from these projections. We will see in the next few sections how the problem can further be reduced to a problem of solving a set of simultaneous linear equations and later,in the next chapter, we will formulate the problem in the manner relevent to our discussion.

2.3.3 Iterative Schemes

In the iterative methods a_{ij} are continually reconstructed in the hope of obtaining a better approximation of the true values on each successive iteration. Some algorithm is developed to calculate the reconstructs a_{ij} from $b_{\Theta i}$. We start with assumed values of the activities a_{ij} and update these values using the projection values $b_{\Theta i}$. The values of a_{ij} obtained above are projected again onto lines at different angles to produce a set of reconstructed projection values $b_{\Theta i}$'s. a_{ij}'s are then recomputed according to some function of $b_{\Theta i}$ and $b'_{\Theta i}$ to give the values for the next iteration. The process is terminated when satisfactory reconstruction has been obtained.

2.4 The Reconstruction Problem as a Set of Simultaneous Equations

The reconstruction space is represented by a basis set which subdivides it into nxn nonoverlapping subregions of similar nature. The unknown density distribution is approximated by the values assigned to each element by the reconstruction algorithm.

Each projection which is finite in extent is also divided into nonoverlapping elements the maximum size of which is dictated by the presumed spatial resolution in the projection.

A projection consists of bands of parallel radiation through the density distribution at a known angle θ_k. Each band is known as a ray and covers some region δ of the distribution. The projected elements are b_j's (the information about the angle has been omitted for ease in writing and $b_{\theta j}$ is written simply as b_j) and corresponding to each b_j there is a δ_j depending on the path of the radiation (Fig. 2.1). b_j is a function of the sum of the activities of all the points lying in the corresponding region δ_j. The projected elements b_j, $j = 1, \ldots n$ for any θ_k form a projection at that angle. If we take projections at p different angles (θ_k, $k = 1, \ldots p$) then we have b_j, $j = 1, \ldots np$ projection elements in all, each having a corresponding δ_j.

If we denote our reconstruction space by S and $\bar{x}(r)$ as our unknown density at point $\bar{r}$ in S then the integration of all such densities in any region δ_j will yield approximately

the measured value of the projection element b_j. Mathematically speaking

$$\int_{\delta_j} x(\bar{r})\, d\bar{r} = b_j \qquad j = 1, \ldots np \tag{2.2}$$

The above equation is the starting point of all reconstruction methods using projections.

If we denote s_i as the ith element of S then we have s_i, $i = 1, \ldots n^2$ elements. Any section δ_j will contain a number of such regions (or parts thereof). Then the integral in (2.2) over δ_j can be broken down into summation of a number of integrals over the regions s_i which lie in δ_j. Thus,

$\int_{\delta_j} x(\bar{r})\, d\bar{r} = b_j$ can be written as

$$\sum_{i=1}^{n^2} \int_{\delta_j \cap s_i} x(\bar{r})\, d\bar{r} = b_j \qquad \begin{matrix} j = 1, \ldots np \\ i = 1, \ldots n^2 \end{matrix} \tag{2.3}$$

This can be arrived at by the definition of a projection element, we gave earlier, as the sum of all the densities associated with the corresponding ray. $\delta_j \cap s_i$ isolates those cells which contribute to the ray intercepting region δ_j of S and having the projection element b_j.

To start with, the density $x(\bar{r})$ is unknown and any algorithm designed for reconstructing the density distribution will aim at approximating the unknown density $x(\bar{r})$ by assigning an

estimate x_i of its value to each region s_i. The best estimate would result when x_i is the average value of $x(\bar{r})$ over the subregion s_i.

$$x_i = \frac{\int_{s_i} x(\bar{r})\, d\bar{r}}{\int_{s_i} d\bar{r}} \qquad i = 1, \ldots n^2 \qquad (2.4)$$

This ideal outcome cannot be achieved due to limitations on the amount and quality of data as well as the reconstruction algorithm themselves. We can, nevertheless, get so close to this estimate, after making some practical assumptions discussed later, that the reconstructed density distribution comes very much within the limits of acceptability.

We do not know how the function $x(\bar{r})$ varies within any region s_i. Without this knowledge we cannot predict the value of $x(\bar{r})$ in $\delta_j \cap s_i$, the region of the subregion s_i contributing to the projection element b_j. To go around this difficulty we assume that x_i is the average value of $x(\bar{r})$ over s_i. We define a geometric function as

$$a_{ij} = \int_{\delta_j \cap s_i} d\bar{r} \qquad i = 1, \ldots n^2 \qquad (2.5)$$

as the area contributing to the element b_j. Then the integral in (2.2) can be approximated as

$$\int_{\delta_j \cap s_i} x(\bar{r})\, d\bar{r} = a_{ij}\, x_i \qquad (2.6)$$

The equation (2.3) reduces to a set of simultaneous equations :

$$\sum_{i=1}^{n^2} a_{ij} x_i = b_j, \quad \begin{matrix} j = 1, \ldots np \\ i = 1, \ldots n^2 \end{matrix} \qquad (2.7)$$

The above system of equation has the following peculiar characteristics.

(1) All the elements of matrix $\{a_{ij}\}$ are nonnegative.

(2) The size of $\{a_{ij}\}$ is enormous e.g. for n = 512, p = 10 $\{a_{ij}\}$ is of the order of 10^9.

(3) $\{a_{ij}\}$ is quite sparse since $\delta_j \cap s_i = \emptyset$ for most pairs (i,j).

(4) The rank of the matrix $\{a_{ij}\}$ is unknown.

(5) The unknown function $x(\vec{r})$ is ordinarily assumed to be nonnegative so that one desires a solution for which $x_i \geq 0$.

(6) The errors in the data may cause the equations to be inconsistent.

(7) The approximations by which we arrive at (2.7) introduce systematic errors which have yet to be analyzed.

The various questions which might arise at this stage are (1) Is it always possible to reconstruct in the above manner? (2) Is there any limitation on the number of projections required? (3) Is there any limitation on the range of angle θ? The discussion below attempts to answer these questions.

In some of the following lines we shall prove [5] that if the reconstruction space containing a finite object is represented by either nxn squares or nxn points, then n projections within an arbitrarily small range of angles suffice to reconstruct the space exactly provided only that these are also known exactly.

The Fig. 2.3 shows a grid of nxn sampling points used for reconstruction of some distribution. We start with defining a ray as a band of finite width across the reconstruction array. All the rays of a projection are parallel to each other, as mentioned earlier.

The points are represented by an ordered pair (i,j) where i denotes the column number and j the row number.

The first ray of the first projection is chosen such that its lower edge passes through a point F_1 on the line through the second row and also between the points (1,1) and (2,1) (see Fig. 2.3). This ray now has all the information about the point (1,1) and can thus be used to determine the activity associated with this point $a_{1,1}$. Now choose the first ray of the second projection in a similar manner and constrain its lower edge to pass through F_1 and also between the points (2,1) and (3,1). This ray will now include the information about the activities $a_{1,1}$ and $a_{2,1}$ at (1,1) and (2,1) respectively. But the activity at (1,1) has already been determined so we can readily determine the activity $a_{2,1}$ associated with the point

(2,1). By induction we may conclude that n rays (all the first rays of n projections) so defined will uniquely determine $a_{1,1}$ through $a_{n,1}$. The second ray of each projection is similarly defined except that it passes through the point F_2. The rays will intercept points of the first row along with the points of the second row. But the first row has been completely determined so the activities associated with the points in the second row can also be determined. In this manner we can define n points F_j, j = 1, ... n which will give us n rays for each of the n projections, these n projections comprising n rays each suffice to determine an nxn picture uniquely.

Another way of discretizing an analog density distribution is by sampling it with a square grid. In this case the density distribution is broken up into nxn square cells of same size (Fig. 2.4). Each square, in this case, is assumed to have a uniform density inside. In this the points F_1, F_2, ... F_n are chosen as shown. The cells are denoted in space by the set (i,j) where i denotes the column and j denotes the row number.

The first ray of the first projection passes through the point F_1 and top right hand corner of the cell (1,1). This ray in doing so intercepts a known fraction of the square (1,1). From this information the density associated with this cell can be determined. The first ray of the second projection passes through F_1 and the top right hand corner of the square

(2,1). This ray intercepts a known fraction of the square (1,1) and a known fraction of the square (2,1). Then density of square (1,1) has already been determined and by the information provided by this ray we can determine the density associated with the square (2,1). Similarly we can define first ray of the projections 3,4, ... n which will successively give the densities associated with the squares (3,1), (4,1) ... (n,1). Now point F_2 can be taken and from this we can pass the second rays of all the projections which also pass through the top right hand corner of squares (1,2), (2,2), ... (n,2). This determines the activities of the cells (1,2), (2,2), ... (n,2) respectively. Similarly we can take points F_3, ... F_n and get n rays of n projections which will together determine the activities associated with all of the nxn cells of the array.

In our study we have assumed the sampling by a grid of n^2 cells such that the density associated with each of the cell is concentrated at its centre instead of being uniformly distributed all over it. This makes the formulation of the reconstruction algorithm simpler. We still have a unique solution if n rays of n projections are known.

2.5 Limits on the coarseness of Division of S-the Factor Determining the Size of $\{a_{ij}\}$

This problem is taken up in a slightly different context in the next chapter where we show how the size of the sampling

grid can affect the picture. With coarser grid a circle is shown to have been converted into a square or a cross on discretization. As the fineness of the grid is increased the picture acquires finer details and more details will be correspondingly retained in the reconstruction.

On the basis of the above argument one is convinced that the division can be made infinitely fine to get the exact reconstruction. But since the size of the matrix $\{a_{ij}\}$ depends directly on the number of subregions s_i in S, the division should be as coarse as possible to restrict the size of s to make computations easier. Again we cannot go on making the division coarser since the size of the projection element b_j puts a limit on it. How b_j restricts the degree of coarseness can be seen from the following argument.

Let c_i be the centroid of the subregion s_i such that $c_i \in s_i$ (convexity). We define a matrix $\{u_{ij}\}$ such that

$$u_{ij} = \begin{cases} 1 & \text{if } c_i \in \delta_j \cap s_i \\ 0 & \text{if not} \end{cases} \qquad (2.8)$$

The product sum $\sum a_{ij} x_i$ gave the sum of the densities assigned to all the elements in the region δ_j. We can approximate the product $a_{ij} x_i$ by $u_{ij} x_i$ so that

$$\sum_{i=1}^{n^2} u_{ij} x_i = b_j \quad , \; j = 1, 2 \ldots np \qquad (2.9)$$

Summing up what we have done till now since we started with the equation (2.2), we have, step by step, approximated this integral by the sum of the estimates at a finite set of points within δ_j. For a given passage δ_j, the number of points it intercepts

$$N_j = \sum_{i=1}^{n^2} u_{ij} \qquad j = 1, \ldots np \qquad (2.10)$$

must be nonzero. This limits the coarseness of the division of **S**.

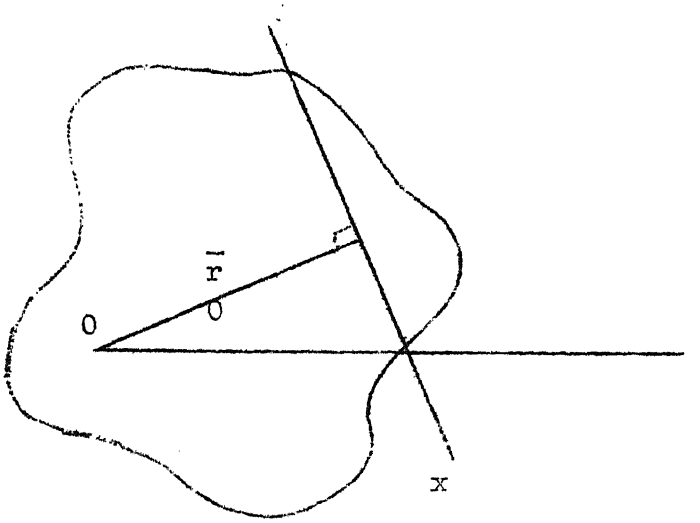


FIG 2.6

CHAPTER 3

DIGITAL RECONSTRUCTION FROM PROJECTIONS

3.1 Introduction

In the second chapter we developed the theoretical basis for our problem of reconstruction. We discussed as to how the projections can be taken and used for reconstructing images in two dimensions. It was mentioned that some sort of sampling is to be done. In this chapter we look into digitization more critically outlining the method we are using, viz., sampling with a square grid. We also show how pictures can be represented in the form of matrices. Once the question of digitization has been settled we develop the problem on hand, in the form in which it is going to be solved.

For convenience we define our analog picture again as being a real valued, nonnegative function, $f(x,y)$, of two real variables x and y. The value of this function at a point is the grey level of the picture at that point (28). It is further assumed that this function is well behaved, i.e. integrable, Fourier transformable etc. When colour is involved the function should be considered as vector valued or several functions should be used. We shall confine our study to pictures with varying intensities of grey level only.

We now look into digitization of this analog picture so that we can express our problem in a convenient form

3.2 The Digital Picture

To achieve representation of continuous space in a discrete form many truncated basis sets have been used to represent the higher dimensional space in reconstruction problem [23].

In our case we have chosen the basis set which divides the given region into a finite number of non-overlapping elements or subregions. We make an assumption that whole of the density of any such subregion is concentrated at the centre of that region. We can do away with this assumption, and not fruitlessly, at the cost of computational time and money. The assumption which we have made gives practically acceptable results with comparatively lesser (still large) computations.

We sample our picture in the following manner : The picture is considered to be lying within a suitable square grid in such a way that all the desired details are encompassed (Fig. 3.2). The grid contains squares of equal size. The density of each of these squares or cell is taken to be concentrated at the centre. These values are taken to be the sample values.

As image processing literature confirms, when sampling of this type is used it is assumed that there is no abrupt change in the grey level of the picture. The levels change in a gradual manner. This allows us to attach one density to each cell of the grid. And this density, as mentioned above, is assumed to be concentrated at the centre of the cell.

The above mentioned method has its implications. It will be much easier to illustrate the difficulty with an example. Consider the picture,to be discretized,to be a circle of grey level one embedded in a back-ground of grey level 0. Following our sampling method we enclose this picture by a grid of size say 4x4 16 cells (Fig. 3.2). Now we have to attach one density value to one cell. The cells which have only one density can be given that value but there is an ambiguity when the cell has two or more grey levels. In the figure cells 6,7,10,11 have their greater part ($\geq$ 50%) filled with grey level 1 so they will be assigned this value. Fig. 3.3 thus gives the picture we are actually going to use for processing. The circle has been converted into a square.

It will be immature to jump to conclusions rendering this method useless. Let us increase the number of cells in the grid keeping the picture area to be same. We make the grid size 8x8 (64 cells). Using the above method of

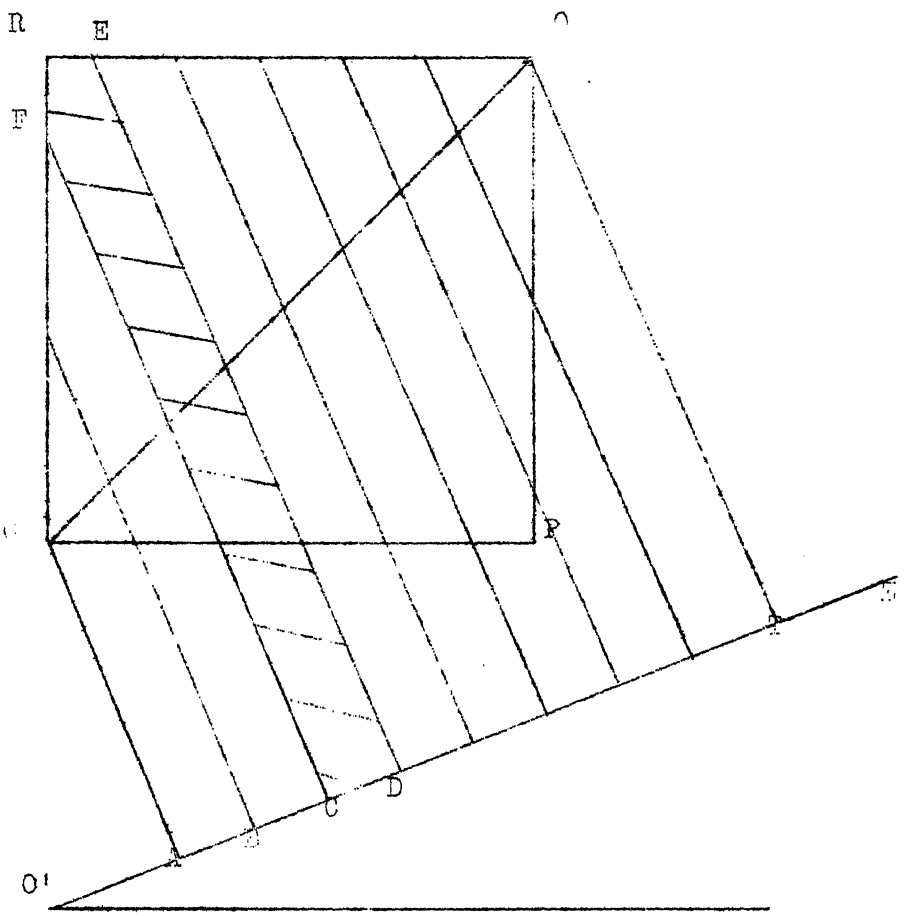


Fig. 3.1 Raysums and projection elements

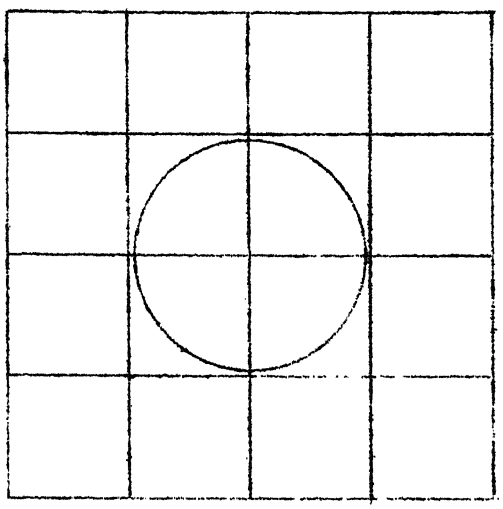


Fig. 3.2 Figure of a circle within a square grid

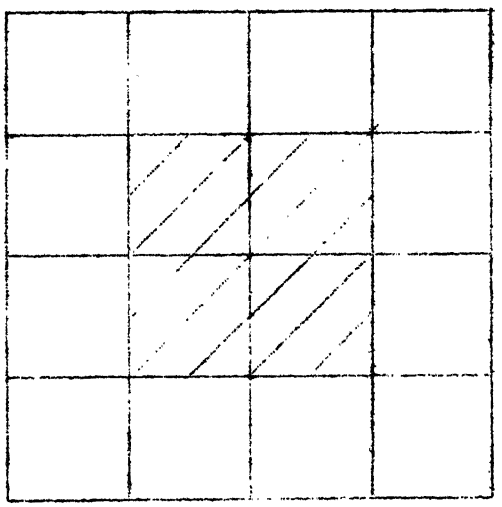


Fig. 3.3 Discretization with a 4x4 grid

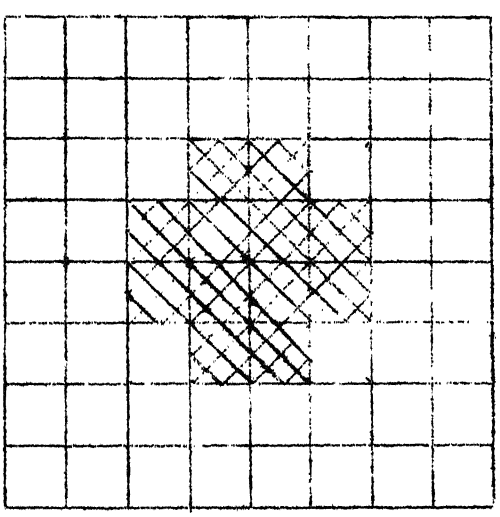


Fig. 3.4 Discretization with a 8x8 grid

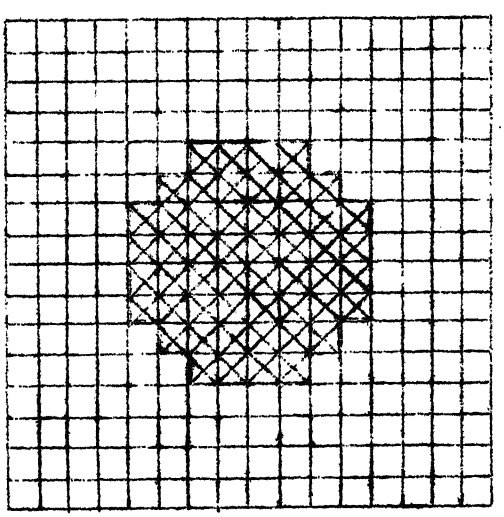


Fig. 3.5 Discretization with a 16x16 grid

assigning values we see that our picture now changes to a cross, not at all acceptable as a circle. But when we make the grid denser i.e. increase the size to 16x16 (256 cells) or more,the circle begins to take shape (Fig. 3.5). Thus a proper grid is to be chosen depending on details of the picture we want to bring out. We can write the cell values in the form of a matrix. A nxn grid leads to a nxn matrix. We observe that finer the sampling grid the greater is the resolution.

Thus a digital picture is defined by specifying a nxn matrix of grey levels. An element of this matrix is called a picture element or pixel.

The unknown picture is approximated by values assigned to each element by the reconstruction algorithm.

3.3 The Actual Problem

We now have a picture digitized into NxN array of distinct cells which are each a part of the whole picture.

After digitization the picture is ready for taking projections. Obtaining projections at any angle consists in dividing the picture space into non-overlapping bands of finite, equal or varying widths at that angle. All the cells whose centres lie within any particular ray contribute to the raysum of that particular ray. The set of

values of the raysums (projection elements) then gives the projection at that angle (P_θ). Each projection thus has a finite number of non-overlapping elements b_i's. The maximum size of the projection element should be dictated by the presumed spatial resolution in the projection.

We will consider projections at angles $\theta_1 \ldots \theta_p$, where p is the number of projections to be taken. Any projection gives an N element vector e.g. b_k is the N element vector corresponding to the projection at angle θ_k.

We may write $b_k = (b_{k,1} \ldots b_{k,N})^T$

for each of the p projections we have such N element vector. If all the p projections are considered at the same time then we can put all the b_k's, k = 1, ... p., together to form a Np dimensional vector.

$$\underline{B} = (b_{1,1}, \ldots b_{1,N}, b_{2,1}, \ldots b_{2,N} \ldots b_{p,1}, \ldots b_{p,N})^T$$

by defining $b_{k,l} = b_{(k-1)N+l}$ $\quad k = 1, \ldots p$ $\quad l = 1, \ldots N$

We get

$$\underline{B} = (b_1, \ldots b_N, \ldots b_{2N}, \ldots b_{(p-1)N+1} \ldots b_{Np})^T$$

b_j corresponds to the raysum of the ray spanning region δ_j in S(ref. Fig. 2.1) e.g. b_3 corresponds to the raysum of the ray spanning region CDEF in OPQR (Fig. 3.1).

$b_1, b_2 \ldots b_{Np}$ can referred to as point projections. These can be related to the vector x^* as follows

$$b_{(k-1)N+l} = \sum_{j \in D_{(k-1)N+l}} x_j^* \qquad \begin{matrix} k = 1, \ldots p \\ l = 1, \ldots N \end{matrix}$$

The set D_i is made up of all the cells which contribute to the projection b_i.

Example : $b_{(k-1)N+3} = y_{k,3}$ and the corresponding set $D_{(k-1)N+3}$ is made up of all the cells which lie in the strip CDEF.

To get the set D_i which contributes to the projection element b_i we consider the following :

Consider a cell (i,j) with center as $(i-\frac{1}{2}, j-\frac{1}{2})$. The intensity of the cell (i,j) (cell $(i-1)N+j$)) can be written as $x^*_{(i-1)N+j}$.

In the Fig. 3.1 OΘ is the main diagonal and ΔT is its projection on the axis O'z at an angle θ_k to the horizon, so we have

$$\Delta T = N \sqrt{2} \cos(45 - \theta_k)$$

With the choice of constant ray width we divide ΔT into NN (NN $\geq$ N) equal parts. In the figure AB = BC = CD ... ST = $\sqrt{2} \cos(45° - \theta_k)$ and NN = N. If the centre of the cell (i,j) falls within ABOG then cell (i,j) contributes to $b_{k,1}$ or if it comes within BCEG then it will contribute

to the projection element $b_{k,2}$ and so on. The projection of the center of the cell (i,j) on the axis O'Z (at an angle θ_k) can be written as

$$\left[(i-\tfrac{1}{2})^2 + (j-\tfrac{1}{2})^2\right]^{\frac{1}{2}} \cos(q_{ij} - \theta_k)$$

where

$$q_{ij} = \tan^{-1}\left[(j-\tfrac{1}{2})/(i-\tfrac{1}{2})\right]$$

Then $D_{(k-1)N+1} = \{(i-1)N+j \; ; \quad 1 \leq i, \quad j \leq N\}$

and (i,j) obey

$$(1-1)\sqrt{2}\cos(45-\theta_k) \leq \left[(i-\tfrac{1}{2})^2 + (j-\tfrac{1}{2})^2\right]^{\frac{1}{2}}$$

$$* \cos(q_{ij}-\theta_k) < 1\sqrt{2}\cos(45^\circ - \theta_k)$$

After achieving familiarity with the mode of discretizing the picture and obtaining the projections we can find the matrix [A] which contains information about the elements contributing to various projection elements. To illustrate the construction of matrix [A] from NxN picture matrix we take a few simple examples : For a 4x4 matrix the projection matrices for projections at 0°, 45°, 90° are shown below.

$$A(0) = \begin{matrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix}$$

$$A(45°) = \begin{matrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

$$A(90°) = \begin{matrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

From the above matrices we can make out that there are rows corresponding to each of the ray of the projection. And the elements which fall within that ray are represented by 1's in the corresponding row of the matrix. The rest of the elements of that row are zeros.

For example, the element b_1 (first element of the projection at 0°) gets contribution of all the first four row elements x_1, x_2, x_3, x_4 and no other elements. So the first four elements of the first row of $\underline{A}(0)$ are 1's and the rest are 0's.

With the above background and recollecting the Section 2.4 we can define our problem more precisely as follows :

Given : The projection vector $\underline{B}$ (= b_1, ... b_{Np}) corresponding to the projection angles θ_1, ... θ_p.

Derivable Information : As we know the angles of projection we can find matrix $\underline{A}'$ for each θ_i, i = 1, ... p and combine them to get the matrix $\underline{A}$.

Problem : With the above information and the given data we can form a set of equations

$$\underline{A}_{\theta i} X = \underline{B} \qquad i = 1, \ldots p$$

The solution X, which has to be obtained using a projection algorithm, will give an estimate of the picture of whose b_i's are the projection elements. Here it has been assumed that we already have the analog picture and we are discretizing it, taking projections and reconstructing the same picture again. This has been done for the purpose of testing our algorithms. In practice we have a number of projections of a real life object obtained by using X rays or some suitable radiations. The problem basically remains the same.

CHAPTER 4

PROJECTION ALGORITHM (PRA) AND ORTHOGONALIZED PROJECTION ALGORITHM (OPRA)

In this chapter we will deal with the iterative scheme we have chosen for our study. The projection algorithm (PRA) was found to be most suitable for our study since it allows incorporation of apriori data, it is applicable to space varying as well as space invarient problems; it is computationally less cumbersome and above all it is guaranteed to converge. Besides, like all iterative techniques PRA has a way of getting implemented with greater ease as compared to the other contemporary techniques. After discussing the projection algorithm, we present a scheme of orthogonalization of the projection equations which contributes to its faster convergence. Binary orthogonalization, in particular, has been discussed in some detail.

4.1 The Projection Algorithm (PRA)

We saw in the last chapter that our problem of reconstruction can be reduced to a problem of solving a set of simultaneous linear equations iteratively in hope of getting better and better solution till the process converges. PRA does just this.

The projection method for solving a system of equations was proposed by Kacmarz [30] in late thirtees. Kacmarz did not prove the guaranteed convergence of this method. Tanabe [31] reviewed this method of solving linear equations in early seventies and also took up the question of convergence. Convergence was established by Tanabe in a complex linear space. Such generality is not required for our problem and we will assume a real linear space in our discussion.

Once the theoretical concepts were available PRA was applied in the form of ART by Gordon, Bender and Herman [5] Gordon and Herman [23], Gordon [21] to the problem of reconstruction of images from their projections. The method was found to be quite successful for images with varying grey levels. With 20 iterations of the proposed algorithm they were able to reconstruct the picture of 'Judy' quite satisfactorily. [5].

The system of equation that has to be solved is :

$$\underline{A}\,\underline{X} = \underline{B}$$

where $\underline{X}$ is M dimensional column vector where M is the size of the picture array (N^2 for an NxN picture). $\underline{B}$ is the projection vector, an NP_p dimensional column vector where p is the number of projections and N is the number of projection elements.

$\underline{A}$ is $N_p x N^2$ dimensional projection matrix. Thus we can say $\underline{X} \in V^M$, $\in B \in V^{Np}$ where V^M and V^{Np} stand for M and N_p dimensional real vector spaces of column vectors.

An assumption will be made at this juncture which will hold throughout this discussion. All the elements of $\underline{X}$, $\underline{B}$ and $\underline{A}$ will be assumed to be non negative numbers. If in the course of solving the system we get negative numbers they will be set to zero before proceeding further.

A mapping f_j $(V^M \rightarrow V^M)$ is defined which maps a M dimensional vector into another M dimensional vector.

$$f_j(x) = \underline{x} - \frac{(\underline{x},\underline{a}_j) - b_j}{(\underline{a}_j,\underline{a}_j)} a_j$$

$$j = 1, \ldots Np$$

a_j is the jth column vector of A^T and $(y.3) = z^T y$.
Another mapping F $(V^{M+Np} \rightarrow V^M)$ is defined as follows

$$F(i,x) = f_1, f_2 \ldots f_{N_p}(x)$$

$\underline{x}$ is an M dimensional vector

We can write this product of mappings as

$$F(i,x) = f_1(f_2(f_3(\ldots f_{N_p}(x)))$$

These can be applied to our assumed solution vector $\underline{x}$ to yield iteratively the desired solution.

If we begin with an assumed solution vector $\underline{x}^o$ then $F(i,\underline{x}^o)$ will give an approximation $\underline{x}'$ of the solution x. Proceeding similarly we can get a sequence of vectors $\{x^k\}$, k = 0,1,2, ... defined by the recursion

$\underline{x}^{k+1} = F(i,\underline{x}^k)$ where $\underline{x}^o$ is the arbitrary initial vector. Tanabe [31] has proved that the sequence $\underline{x}^k$ converges for arbitrary $\underline{x}^o$ to the solution if it exists and is unique or to a solution $\hat{\underline{x}}$ for which $||\underline{x}^o - \hat{\underline{x}}||$ is minimum.

The physical interpretation of the mapping f_j and F is as follows: The operator $f_j : V^M \rightarrow V^M$ projects orthogonally any vector in space V^M onto the hyperplane defined by $(\underline{x}, \underline{a}_j) = b_j$.

The operator $F: V^{M+Np} \rightarrow V^M$ projects orthogonally any vector successively onto the hyperplanes defined by $(\underline{x}, a_j) = b_j$, j = 1, ... Np in the order j = Np, Np-1 , ... 3,2,1. One complete operation by F in any order is called an iteration. In terms of the computer algorithm of PRA, one iteration updates all the elements of $\underline{x}(x^o)$ giving x' and bring this vector closer to the actual solution. The operation is repeated and we get, successively, x^o, x^1, x^2, ... x^n till the vector $\underline{x}$ converges to the solution x, if it exists, or to a vector $\underline{x}_A$ (acceptable solution) for which $||\underline{x}^o - \underline{x}_A||$ is minimum. The iterations,

in practice, may be terminated by human interference after a visibly best solution has been obtained.

The method seems to be particularly attractive because of its guaranteed convergence. One iteration of this algorithm requires approximately $2M * Np$ ($= 2 N^2 * Np = 2 N^3 p$) multiplications and as many additions. Here N^2 is the size of the picture and p is the total number of projections giving Np equations. $Np << N^2$ in practical cases so the number of computations required, though much less than those required in the direct matrix inversion method, are still enormous for most practical images. Some way of reducing the total number of computations required (and hence computation time) in reconstructing an image is desirable.

A particular scheme suggested by Ramakrishna [33] in the context of the restoration problem, has been applied to the current problem which by accelerating the speed of convergence yields a reduction in overall reconstruction time. This scheme has been tested with a number of examples using binary and multidensity digital picture data. It has been confirmed that the method achieves faster convergence in almost all the cases though the gains vary with the individual pictures.

4.2 Aspects of Orthogonalization

The algorithm we propose involves some degree of orthogonalization and so a brief discussion of the same is in order.

We recall that a set X is called a linear space over a field K if the following conditions are satisfied: X is an abelian group written additively. A scalar multiplication is defined in the following manner - to every element $x \in X$ and each $\alpha \in k$ there is associated an element in X denoted by αx.

The following properties also hold good :

$\alpha (x+y) = \alpha x + \alpha y (\alpha \in k,\ x,y \in X)$

$(\alpha+\beta)x = \alpha x + \beta x (\alpha, \beta \in k,\ x \in X)$

$(\alpha\beta)x = \alpha(\beta x) (\alpha, \beta \in k,\ x \in X)$

i.e. $1.x. = x$ (1 is the unit element of field k)

Consider a linear space with a system of vectors in that space. This space will be an N dimensional space if it has atleast N independent vectors and each of them has a unique representation $x = \sum_{j=1}^{N} \alpha_j y_j$. The seminorm of any vector in this space gives a kind of length for that vector. If the above system of vectors reduces to a single seminorm, the corresponding linear space is called normed linear space.

With this background we state Schmidt's orthogonalization theorem [36]. The theorem states that given a finite or countably finite sequence $\{x_j\}$ of linearly independent vectors of a pre Hilbert space (see def) X. Then we can construct an orthonormal set having the same cardinal numbers as the set $\{x_j\}$ and spanning the same linear space as $\{x_j\}$.

4.3 PRA with Orthogonalization (OPRA)

Each of the equation in the system $\underline{A}\ \underline{X} = \underline{B}$ can be considered to be a hyperplane in a M dimensional space. If a unique solution exists then these planes intersect at a common point which is the solution. Tanabe, as we have said before, has established convergence and the convergence is assured even if the equations are ordered in any arbitrary manner. Though convergence is independent of the ordering of equations the speed is very much dependent on it and this fact has been exploited in the algorithms used. We start with an arbitrary point in the M dimensional space as the solution. Every projection of this point on a given hyperplane results in the movement of that point towards the intersection of the hyperplanes if the there is any, else towards a point which gives the minimum norm solution. The point is successively projected onto various hyperplanes (corresponding to different equations)

and points nearer and nearer to the solution are achieved. After repeated projection of the point, it captures the solution point and the process is terminated. If there is no point of intersection then successive projection of the initial guess onto the hyplanes causes it to move to a point so that the norm of the difference between the latter and the initial guess is the minimum.

Motion is accelerated if the planes are wide apart, the greater the angle between any two planes, the faster is the motion towards the point of intersection or to the minimum norm solution. The ideal case will be when all the planes are orthogonal to each other since in this case convergence will be achieved in a single iteration for any ordering of equations.

This sort of complete orthogonalization can, theoretically, be achieved by the Gram-Schmidt orthogonalization process [33]. However, practically such an endeavour will call for enormous computational effort and even after that we might end up with a system completely different from the original one due to computational round offs. This will render the reconstruction completely useless as the reconstructed image will, in the worst case, have no similarily with the original one.

Failure or unsuitability of complete orthogonalization should not be taken to mean that orthogonalization is not at all useful for reconstruction procedures. We can resort to partial orthogonalization, i.e. the more practical m-ary orthogonalization ($m << Np$). In particular, binary ($m = 2$) orthogonalization is comparatively less cumbersome to implement and gives a saving of upto 50% on the computational time.

4.4 Binary Orthogonalization

In this case instead of making each of the planes orthogonal to all other planes we do pairwise orthogonalization. This consists in making any two successive hyperplanes orthogonal to each other. This is practically feasible. The following algorithm has been used in principle.

i) Take the first equation to be fixed and orthogonalize the second equation w.r.t. the first equation,

ii) Orthogonalize the third equation w.r.t. the second equation obtained in (i).

iii) Continue till the Npth equation has been orthogonalized w.r.t. the (Np - 1)th equation which had already been orthogonalized w.r.t. the orthogonalized (Np-2)th equation

iv) Now orthogonalize the first equation w.r.t. the Npth equation.

Mathematically we may state that :

$$\underline{a}^{o}_{j+1} = \underline{a}_{j+1} - \frac{(\underline{a}_{j+1}, \underline{a}^{o}_{j})}{(\underline{a}^{o}_{j}, \underline{a}^{o}_{j})} \underline{a}^{o}_{j}$$

$$b^{o}_{j+1} = b_{j+1} - \frac{(\underline{a}_{j+1}, \underline{a}^{o}_{j})}{(\underline{a}^{o}_{j}, \underline{a}^{o}_{j})} b^{o}_{j}$$

$$\underline{a}^{o}_{1} = \underline{a}_{1} - \frac{(\underline{a}_{1}, \underline{a}^{o}_{Np})}{(\underline{a}^{o}_{Np}, \underline{a}^{o}_{Np})} \underline{a}^{o}_{Np}$$

$$b^{o}_{1} = b_{1} - \frac{(\underline{a}_{1}, \underline{a}^{o}_{Np})}{(\underline{a}^{o}_{Np}, \underline{a}^{o}_{Np})} b^{o}_{Np}$$

$j = 1,2, \ldots$. Np, the vectors superscripted with o denote the vectors which have been modified after orthogonalization except in the first step when second equation is orthogonalized w.r.t. the fixed first equation.

The number of computations required are $2MN$ $(= 2N^2 * Np = 2N^3 p)$ multiplications and an equal number of additions. This is same as the computations required for one iteration of PRA.

This method has been intermingled with the projection algorithm in a manner such that only two equations are taken up into the computer core memory at any time, second is orthogonalized with respect to the first and the former is used for modification of the solution vector. The second equation is now shifted into the place of the first and the third one is

brought into the original place of the second and the operation is repeated till all the equations have been exhausted. This completes one iteration of the algorithm.

Notationally :

A1
A2 two arrays to store the two equations

(A) denotes the contents of A array

j = 0

Start : j = j+1

A1 ← jth equation

Increment : j = j+1

A2 ← jth equation

Orthogonalize (A2) w.r.t. (A1)

(A1) ← (A2)

Use (A1) for modification of x

IF (j = Np) STOP

Else (Go to increment)

This implements OPRA.

Many of the equations in the system might be very nearly linearly dependent. Orthogonalization tends to seperate them, making them linearly independent and allows OPRA to have greater speed of convergence.

PRA in combination with binary orthogonalization was implemented and the results proved our point. In most of the

test cases we achieved 10 - 50%saving on the computational time. The results are presented in the next chapter.

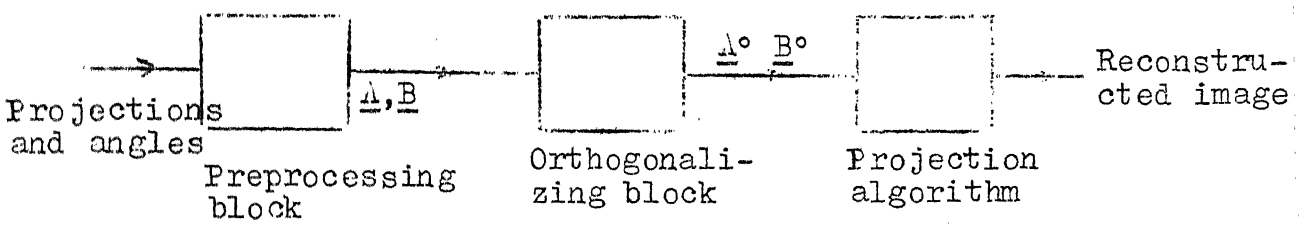


Fig 4.1 Reconstruction model

CHAPTER 5

RESULTS AND CONCLUSION

In this chapter we present the results obtained for the projection algorithm (PRA) and orthogonalized projection algorithm (OPRA). Various error measure have been compared, for the two cases, and the results evaluated. The saving in computational time and cost has been mentioned in each case. Reconstructed pictures using the line printer imagery have been included for visual appraisal. Curves have been plotted with relative errors, after multiplication by a constant factor. The tables have entries of relative errors, rather than absolute errors, for a more effective comparison.

1. Reconstruction of Binary Pattern

Figure 5.1(a) shows a 32x32 picture of an elephant. This figure was used as a test pattern for the algorithms discussed in Chapter 4. Results obtained for OPRA are being compared with those obtained for the PRA. Various numbers of projections were used and the results compared.

With 10 Projections :

Figures 5.2(a) and (b) shows some reconstructions for 10 projections. The table below gives the RMS error and the relative delta for a number of iterations.

No. of iterations	RMS Error		DELTA	
	PRA	OPRA	PRA	OPRA
1	5588	4565	2360	1928
2	3993	3432	1686	1449
3	3419	3025	1444	1277
4	3078	2784	1300	1176
5	2867	2630	1211	1111
6	2725	↓	1152	↓
7	2625	Steady	1110	Steady

Table 1.

The curves in Fig. 5.14 show that the RMS error in the case of OPRA is always less than that in the case of PRA. As depicted in Table 1, the relative RMS error in OPRA stabilizes to a particular value in 5 iterations while it takes PRA 7 iterations to reach that level and then only it settles down. The same table shows relative values of error measure DELTA in the two cases for several iterations. OPRA does in 5 iterations what PRA does in 7 iterations and again the errors in OPRA keep below those in PRA for all iterations. This can be confirmed from the curves of figure 5.13.

Number of mismatches between the actual picture and the reconstructed picture is directly related to the visual quality of the reconstructed picture. This measure tells us at how many points the two pictures differ. In calculating the mismatches we compared the reconstructed F matrix with the picture matrix C. While all the entries of the C matrix were either 0 or 1 the entries in F matrices, after a desired number of iterations, ranged from 0.0 to about 1.6. All values above 0.5 were taken to be 1 and all below 0.5 were taken to be 1. The values equal to 0.5 were also considered to be 1. The two matrices were then compared to give the number of mismatches. The PRA settled down to an error of 3 mismatches in 6 iterations while OPRA settles down to 6 mismatches in 3 iterations. Thus convergence is achieved faster in the case of OPRF though point of convergence may be different.

The above observations with 10 projections show that OPRA gives a saving of about 30% - 40% in computational time and cost.

With 4 Projections :

The Table 2 shows the relative root mean square error for four projections for the figure of an elephant. Here we see that the RMS error in OPRA is always less than the RMS error in PRA. It gives in this case an advantage of atleast 2 iterations. See Figure 5.2(c) and (d).

The Number of iteration	RMS		DELTA	
	PRA	OPRA	PRA	OPRA
1	6190	5793	2615	2447
2	5340	4803	2253	2029
3	4795	4412	2025	1864
4	4439	4176	1875	1764
5	4377	4017	1849	1697
6	4314	-	1822	1648
7	4052	-	1712	-
8	-	-	1623	-

Table 2.

The same table show relative DELTA. This again verifies that error is lower in the case of OPRA. PRA reaches to a comparable value two iterations after the OPRA has reached it. The convergence is thus faster in OPRA and we have a saving of around 30% in the computational time and cost.

Mismatches were calculated for 4 projections and a number of iterations. PRA was seen to settle down in 7 iterations while OPRA settles to nearly the same reconstruction in 5 iterations. We again have a saving of about 30%.

2. Multidensity Pictures

a) First we consider the test picture shown in Fig. 5.1(b)

Reconstruction with 10 projections :

To aid visual comparison mismatches were evaluated in the following manner. All densities, in the reconstructed picture, above and equal to 0.5 were taken to be 1.0 and all those below 0.5 were taken to be 0.0. Mismatches were calculated for PRA and OPRA and the results are given in the Table 3 below.

No. of iterations	Mismatches		RMS	
	PRA	OPRA	PRA	OPRA
1	99	97	6694	5628
2	40	40	5059	4304
3	33	32	4509	3924
4	29	27	4275	3751
5	26	25	4136	3646
6	24	22	4034	3570
7	22	17	3951	3509
8	20	17	3883	3458
9	18	17	3824	3415
10	17	17	3773	3378

Table 3.

We see that both the methods start off with about the same number of mismatches but OPRA converges to 17 mismatches in 7 iterations while PRA keeps on converging slowly and reaches that solution in 10 iterations. Fig. 5.3(c), (d) are included for reference. A clear saving of 30% is observed.

The relative mean relative error in the two cases is plotted the Fig. 5.15 which shows that OPRA, has lower error in general. The same holds for the relative RMS in two cases. See Table 3 and Fig. 5.16.

Reconstruction with 8 projections :

Curves in Figure 5.17, shows a plot of mismatches versus iteration. Both PRA and OPRA have same number of mismatches in the initial steps but as more iterations are performed OPRA accelerates to the solution. OPRA reaches 27 mismatches in 9 iterations while PRA reaches the same value in 13 iterations. The iterationwise mismatches are shown in Table 4. Again a saving of about 30% in computational time is thus achieved.

Mismatches	No.of iterations	1	2	3	4	5	6	7	8	9	10	11	12	13
	PRA	118	63	51	41	48	35	33	32	28	28	29	28	27
	OPRA	118	61	52	38	36	35	31	30	27	27	27	27	27

Table 4.

No.of iterations		1	2	3	4	5
RMS	PRA	6079	4543	4238	4030	3914
	OPRA	6025	4487	4201	4006	3898

Table 5.

The relative RMS errors are shown in the Table 5. The observation confirms the results obtained above.

Reconstruction with 6 projections : (Fig. 5.5(c) and (d))

No.of iterations		1	2	3	4	5	6	7	8	9	10	11	12
Mismatches	PRA	130	89	82	78	76	74	70	65	65	63	60	59
	OPRA	128	83	77	75	71	69	68	64	61	59	59	59

Table 6.

No.of iterations		1	3	5	7	8	9	10
RMS error	PRA	7126	5660	5449	5229	5039	5038	4961
	OPRA	7071	5484	5266	5154	5000	4881	4801

Table 7.

The mismatches, in Table 6, show a gain of about 20% and the table of relative RMS error (Table 7) shows that the error in the case of OPRA is always lower, see Fig. 5.18.

Reconstruction with 4 projections :

With 4 projections we get in 3 iterations of OPRA what we get in 5 iterations of OPRA. A saving of 40% is achieved in this case. See Figures 5.6(c) and (d).

From the above discussion we see that for this test pattern, for all sets of projections 4,6,8,10 we achieve faster convergence with OPRA and there is a saving of 20% - 40% in computational time and cost.

b) A three density pattern

In Fig. 5.1(c) we show a pattern having large triangular areas of densities 0.53 and 0.93 and small areas of density 0.0. Reconstruction for various projections are described below.

Reconstruction with 6 projections : (Fig. 5.7)

Mismatches were calculated by fixing some range for the various densities after observing the F matrix for several iterations. Values of density below 0.15 were taken to be 0. Values between 0.15 and 0.67, including the lower limit, were taken to be 0.53 and values above 0.67 were taken to be 0.93.

The mismatches obtained for 6 projections in the two cases are shown below.

No.of iterations		1	2	3	4	5	6	7	8	9	10	11	12	13
Mismatches	PRA	299	155	116	95	81	75	67	62	58	54	53	54	53
	OPRA	285	157	114	96	84	76	71	67	60	57	51	51	51

Table 8.

The mismatches show a behaviour different from those in the test picture 1 (elephant) and test picture 2 (windows). Initially PRA performs better but after a few iterations OPRA catches up and converges to the solution much before PRA. OPRA converges to a solution having approximately 51 mismatches in 11 iterations while PRA takes 14 iterations to settle down to a steady solution. Gain in this case is nearly 21%.

Later the picture was reconstructed using a number of projections 4,5,6,9 to study mean relative error for comparison. The error was plotted for all these projections with a fixed number of iterations (3). The curves are shown in Fig. 5.19. We can see that the error remains low in OPRA. We can also see that increasing the number of projections above 6 is more fruitful in the case of OPRA.

c) Graded square

The picture shown in Fig. 5.1(d) consists of four slices of different densities, namely, 0.0, 0.42, 0.56 and 1.0.

The test picture was reconstructed using a number of projections. Representative results are given below.

Reconstruction with 5 projections :

The test picture was reconstructed using 5 projections (Fig. 5.11) and varying number of iterations. Various errors were calculated. The way the mean relative error varies with the iterations is shown in the Fig. 5.20. At times OPRA performs almost same as PRA or slightly worse but ultimately OPRA gains over PRA and settles down to a tsteady error after 11 iterations while PRA settles down after 14 or 15 iterations. A gain of over 20% in computational time and cost is thus obtained.

RMS error in the two cases have the following trend :

No.of iterations		1	3	5	8	10	12
RMS error	PRA	7316	3115	2866	2737	2693	2664
	OPRA	7057	3085	2866	2737	2688	2656

Table 9.

OPRA starts on a better note but from 5 to 8 iterations its performance is no better than PRA. After 8 iterations again improves over PRA, giving lesser RMS error.

The overall nearness factor, DELTA, has a similar

Reconstruction with 8 projections :

No.of iterations	RMS error		Mean relative error		DELTA	
	PRA	OPRA	PRA	OPRA	PRA	OPRA
1	6264	5763	2250	2241	1679	1603
2	3894	3558	1370	1351	1044	990
3	2886	2638	978	962	774	734
4	2397	2179	805	789	642	606
5	2105	1908	701	685	564	531
6	1906	1726	632	618	511	480
7	1766	1596	586	571	473	444
8	1662	1500	552	538	445	417
9	1582	1427	526	513	424	397
10	1519	1369	506	493	407	381

Table 10.

The above table(and the curve in Fig. 5.21)show the relative RMS error, mean relative error and DELTA in the two cases. Once again in the case of OPRA the error values are always lower. We gain here atleast two iterations in the steady state.

Reconstruction with 11 projections :

With 11 projections the following error values were obtained.

No.of iterations	RMS error		Mean relative error		DELTA	
	PRA	OPRA	PRA	OPRA	PRA	OPRA
1	7217	6484	2329	2327	1934	1804
2	4491	3466	1514	1289	1024	959
3	3735	2836	1266	1063	1001	789
4	3443	2549	1160	951	923	709
5	3303	2387	1106	886	885	664
6	3217	2282	1072	844	862	635
7	3157	2206	1046	814	846	-
8	3113	2147	1026	790	-	-
9	3078	2098	1010	771	-	-
10	3050	2060	997	-	-	-

Table 11.

All the three error criteria have been evaluated and compared. OPRA performs exceptionally well in this case. All the errors, RMS, mean relative error and DELTA keep much below those in PRA. There is a saving of over 50%. The curves can be seen in Fig. 5.22.

For this test picture we see that with 8 projections we are performing better as compared to 11 projections. This can

happen and is a normal phenomenon 26 . This can be explained by observing that more the number of points on the edge of the ray, the greater is the error since the point can be assigned to only one ray while the square, of whose this point is the centroid, might be contributing almost equally to two rays.

The gains with OPRA have so far been stated in the form of % gains. We can get the idea of savings in terms of the units of time and money by the following : 10 minutes of CPU time cost around Rs. 120 on DEC 1090. Thus if PRA takes 10 minutes to reconstruct a picture matrix then a saving of 30% on the computational time will mean a saving of about 3 minutes and in cost of about Rs. 36.

Conclusion :

Fror the discussion above and the conclusions drawn at each step we can say that binary orthogonalization does speed up the algorithm (PRA) under study. The four test pictures that we have taken for our study, provide enough proof of the superiority of OPRA over PRA.

APPENDIX A

A.1 Reduction of Dimensionality of the Problem

A three dimensional object can be considered to be made up of several 2-D slices put together. If, for example, f(x,y,z) is the 3-D object then f(a,y,z) represents a slice at x = a. These 2-D slices can then be reconstructed using the projection algorithms proposed in the study and they can then be stacked up to reproduce the original 3-D object. Similarly an N dimensional object $f(x_1,x_2,\ldots x_n)$ can be reduced to slices of (N-1) dimensions, viz. $f(a,x_2,x_3, \ldots x_n)$. The (N-1) dimension problem can be reduced to (N-2) dimension problem and finally, using such reductions we can get N dimensional problem reduced to a number of 2-D problem.

We favour reduction of dimensionality since it reduces storage requirement and simplifies the algorithms.

A.2 Reduction of Storage Requirements

In the reconstruction problem we have to deal with large matrices. Even a 4x4 grid produces an A matrix of the order Npx16 where Np is the number of projects. For only 4 projections the size becomes 16x16! But we observe that the matrix A has either 0's or 1's and most of the elements are 0s. Such matrices are called sparse matrices. In such

circumstances storing up of whole of the matrix will be unwise. There are a number of standard procedures which can be used for storage and retrieval of such matrices. In one procedure we store the row numbers and column numbers of the 1's. This reduces the storage requirement (as in A(45) Ch.3) upto 50%. Also we observe that there is a sort of circular symmetry in that the last row can be obtained by right shifting the first row a fixed number of times, the last but one row can be obtained by similarly shifting the second row and so on. This means that it is required to store only half the number of rows of A. This presumes that the number of projection elements, in any projection, is even. A saving of 50% is again obtained.

A.3 Compactness of Figures in Digital Pictures

The square of the perimeter of a figure divided by its area, p^2/A, is a classical measure of non compactness of the figure. In plane geometry p is the normal perimeter of the figure and p^2/A takes a value greater than or equal to 4π, 4π being the lower limit which a figure can take only if its a circle. All the other figures are less compact than circles. For example, for a figure of a square on side s, $p = 45$, $A = 5$, $p^2/A = 16 > 4\pi$. But for digital pictures certain squares or octagons can yield smaller values of p^2/A than do the digitized circles. This depends on how the p is measured. But there is no family of shapes

for all of which p^2/A takes on a unique minimum value .

Digital Perimeter

There are two different ways of measuring p of a figure s which require attention.

1) P can be taken to be the area of the border of where the border of s consists of those points of s that have horizontal or vertical neighbours not in s.

2) P can be evaluated by applying the border following algorithm to s **[51] [52]** by counting 1 for each horizontal and vertical move and $\sqrt{2}$ for each diagonal move.

The method 1 will be taken to give perimeter P_1 and the method 2 will be taken to give perimeter P_2.

Ex1

1 1 1

1 1 1

1 1 1

a digital square : for this $P_1 = 8$, $P_2 = 8$ (since all the 8 moves are in horizontal or vertical directions)

Ex.2 Ex.2.

1

1 1 1

1 1 1 1 1

1 1 1

1

a digital diamond : for this $P_1 = 8$, $P_2 = 8\sqrt{2}$ (since in the case of P_2 all the 8 moves are diagonal)

Ex. 1

1

1

1

a figure of width 1 : for this $P_1 = 4$, $P_2 = 6$ (since in the border algorithm we start and end at the same point)

Digital compactness ratios, P^2/A

Consider a regular octagon O(h,k) whose horizontal and vertical sides have h points each and the diagonal sides have d points each.

Then $P_1 = 4(h-1) + 4(d-1)$

$P_2 = 4(h-1) + 4(d-1)\sqrt{2}$

$A = (h+2(d-1))^2 + 4 \cdot \frac{1}{2} (d)(d-1)$

$= h^2 + 4h(d-1) + 2(d-1)(d-2)$

We observe :

1) For a square $d = 1$

So,

$P_1 = 4(h-1) + 4(d-1) = 4(h-1)$

$P_2 = 4(h-1) + 4(d-1)\sqrt{2} = 4(h-1)$

$A = h^2$

Thus $P_1^2/A = P_2^2/A = \frac{16(h-1)^2}{h^2}$

16 for large h

2) For a diamond, h = 1

There, $P_1 = 4(d-1)$

$P_2 = 4(d-1)\sqrt{2}$

$A = 2d(d-1) + 1$

Thus $2P_1^2/A = P_2^2/A = 32(d-1)^2/[2d(d-1) + 1]$

for large d $P_1^2/A \to 8$ and $P_2^2/A \to 16$.

3) For an octagon with h = d

$P_1 = 4(h-1) + 4(d-1) = 8(h-1)$

$P_2 = 4(h-1) + 4(d-1)\sqrt{2} = 4(h-1)(1+\sqrt{2})$

$A = 7h^2 - 10h + 4$

$P_1^2/A = 64(h-1)^2/(7h^2 - 10h + 4)$

for large h $P_1^2/A \simeq 64/7 = 9.14$

$P_2^2/A = 16(3+2\sqrt{2})(h-1)^2/(7h^2-10h+4)$

for large h $P_2^2/A \simeq 16(3+2\sqrt{2})/7 = 13.32$

4) A digital circle of radius r can be defined as a set of points whose distances from the centre are less than $(r+\frac{1}{2})$.

P_1^2/A for such a circle was found to be nearly 10.2.

P_2^2/A was about 13.7.

Thus P_1^2/A is smallest for a diamond and P_2^2/A is smallest for an octagon.

APPENDIX B

B.1 Number of Views Required for Reconstruction

If a reconstructed image is to be uniformly resolved to a resolution d of a completely unsymmetrical object, the numbers of discrete views must be at least [11] [12]

$n = \pi D/d$, where D is the dimension of the object

Thus for a resolution of 2 cms. in imaging a head 20 cm in diameter, we need nearly 32 views.

But here we have assumed that the object is completely unsymmetrical and random but this is not the most representative case. The practical objects of interest, be it any part or organ of the human body or be it a celetial body, there is a great departure from complete randomness. This allows 5 to 20 projections to suffice in usual reconstructions.

B.2 Seperation between Projections - limits on the Angle of Projection

If we take projection at 0° (shown by broken lines) then for the first projection element OB the pixels involved will be x_1, x_2, x_3, x_4.

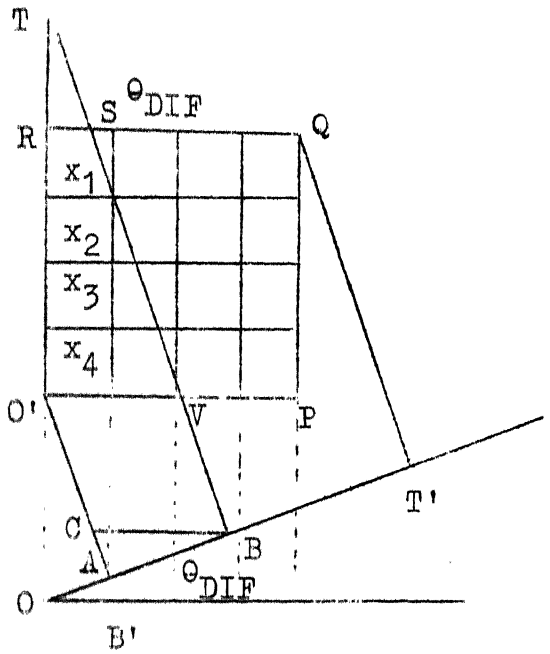


Fig. B.2.1 Discretization of the angles of projection

If we increase the angle slightly then we still have only x_1, x_2, x_3, x_4 contributing to the first projection element. Now we go on increasing the angle till the first ray just misses the pixel x_4. At this instant the angle has been named θ_{DIF}. Now these two projections (at 0° and at θ_{DIF}) will have some different information to convey. The two equations corresponding to the two projection elements OB' and AB will not be totally linearly dependent.

The situation depicted in the figure B.2.1.refers to a difference of one cell width (OV - R5) between the intersected lengths at the point of entering s and the intersected length at the point to leaving (V) the object.

We have to find the angle θ_{DIF}, by which any two

projections must be seperated to carry different informations.

From the right angled triangle VOT

$$\tan \theta_{DIF} = \frac{OV}{OT} \qquad (B.2.1)$$

From the right angled triangle SRT

$$\tan \theta_{DIF} = \frac{RS}{RT} \qquad (B.2.2)$$

From triangle BAC

$$AB = \sqrt{2} \cos(45 - \theta_{DIF}) \quad \text{since} \quad AT = \sqrt{2}\, N \cos(45° - \theta_{DIF})$$

$$BC = \frac{AB}{\cos \theta_{DIF}} = \frac{\sqrt{2} \cos(45° - \theta_{DIF})}{\cos \theta_{DIF}}$$

$$= 1 + \tan \theta_{DIF}$$

$$OV = BC = 1 + \tan \theta_{DIF} \qquad (B.2.3)$$

From (B.2.1) and (B.2.3)

$$OT = \frac{OV}{\tan \theta_{DIF}} = 1 + 1/\tan \theta_{DIF}$$

From (B.2.2) and (B.2.3)

$$RS = RT \tan \theta_{DIF}$$

$$= (OT - OR) \tan \theta_{DIF}$$

$$= OT \tan \theta_{DIF} - N \tan \theta_{DIF}$$

$$= 1 + \tan \theta_{DIF} - N \tan \theta_{DIF}$$

Since OV - RS = 1

$$1 + \tan \theta_{DIF} - 1 - \tan \theta_{DIF} + N \tan \theta_{DIF} = 1$$

$$N \tan \theta_{DIF} = 1$$

$$\theta_{DIF} = \tan^{-1} (1/N)$$

for N = 4, $\theta_{DIF} = 13.5$ so the projections must be spaced atleast 13.5° apart to give useful information.

B.3 Error Measures

The following error measures have been used to compare the algorithms under test.

1. The overall nearness factor DELTA,

$$DELTA = ((1/N^2) \sum_I (\underline{C}(I) - \underline{F}(I)^2)^{\frac{1}{2}} \quad ; I = 1, \ldots N^2$$

2. The Root mean square (RMS) error,

$$RMS = (\sum_I (\underline{C}(I) - \underline{F}(I)^2 / \sum_I (C(I) - AVG)^2)^{\frac{1}{2}}$$

$$I = 1, \ldots N^2 \qquad AVG = (\sum_I C(I))/N^2$$

3. The mean relative error (MRERR)

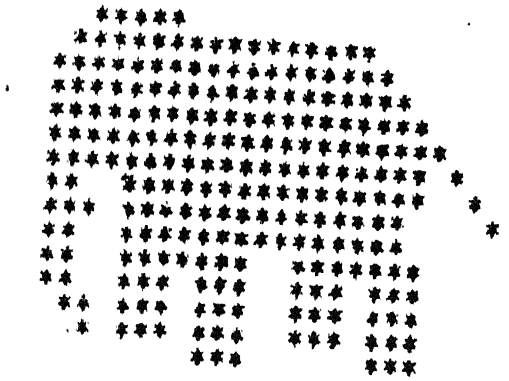
$$MRERR = (\sum_I |\underline{C}(I) - \underline{F}(I)|) \sum_I C(I) \quad ; I = 1, \ldots N^2$$

In all the above definitions C(I) are the picture matrix elements and F(I) are the reconstructed matrix elements.

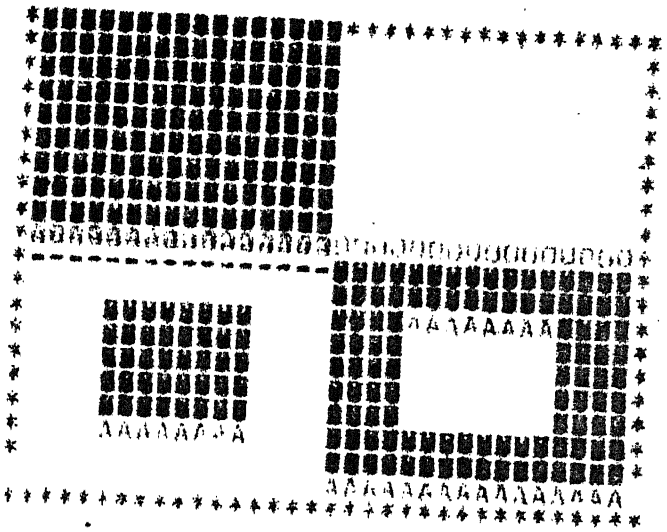
APPENDIX C

The basic code employed, with the approximate density values on a normalized scale, are as follows :

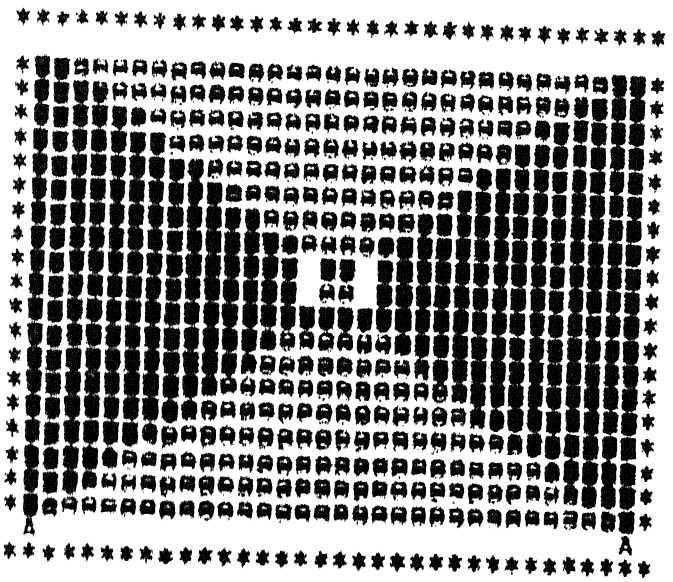
	0.00
-	0.15
=	0.22
+	0.25
)	0.29
1	0.33
Z	0.37
X	0.40
A	0.42
M	0.45
O -	0.53
O =	0.56
O +	0.60
O + ,	0.64
O + , .	0.67
O + , . =	0.79
O X , . -	0.85
O X , . H C	0.89
O X , . H B	0.93
O X , . H B V	0.97
O X , . H B V A	1.00



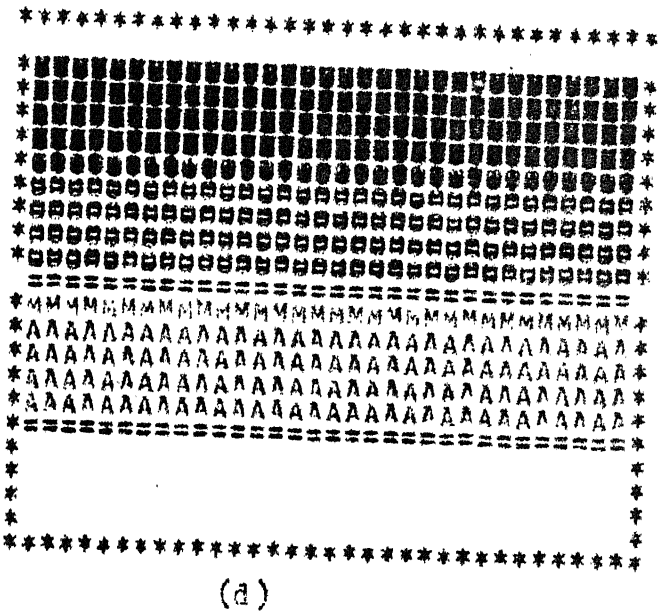
(a)



(b)

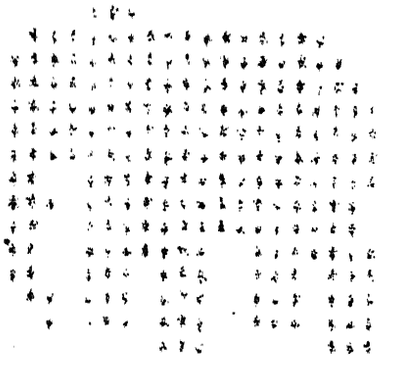


(c)

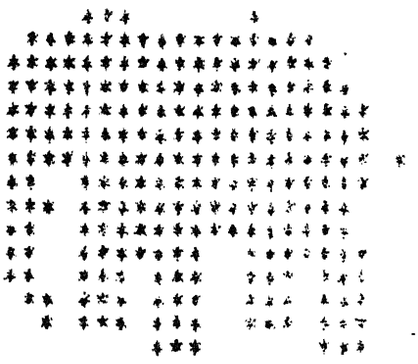


(d)

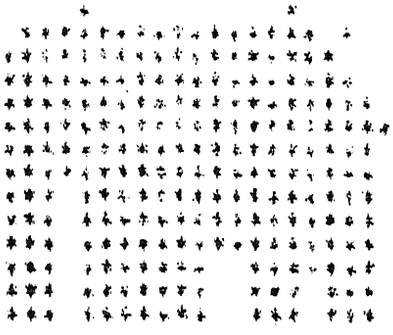
Fig. 5.1 : Original test pictures : (a) Elephant (b) Windows
(c) A 3-density pattern (d) Graded density square



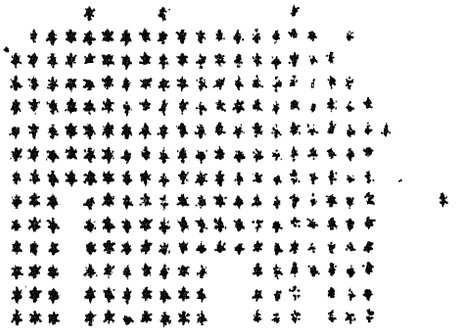
(a)



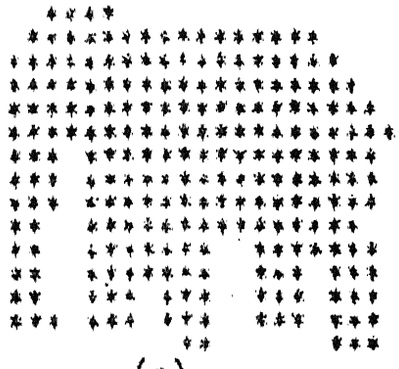
(b)



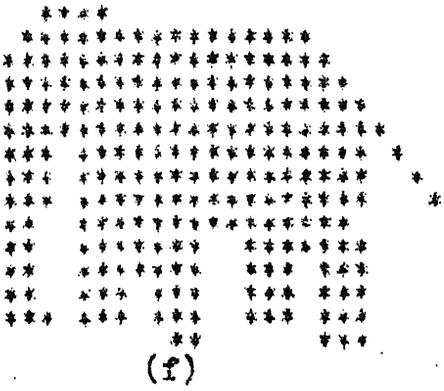
(c)



(d)



(e)



(f)

Fig.5.2 : Reconstruction of the picture of the elephant : NPR=10,NEQ=473, NIT=6, (a) PRA, (b) OPRA, NIT=3; NPR=4,NEQ=189,NIT=7, (c) PRA,(d) OPRA; NPR=8,NEQ NEQ=378,NIT=5, (e)PRA,(f) OPRA, NPR=8,NEQ=378,NIT=5.

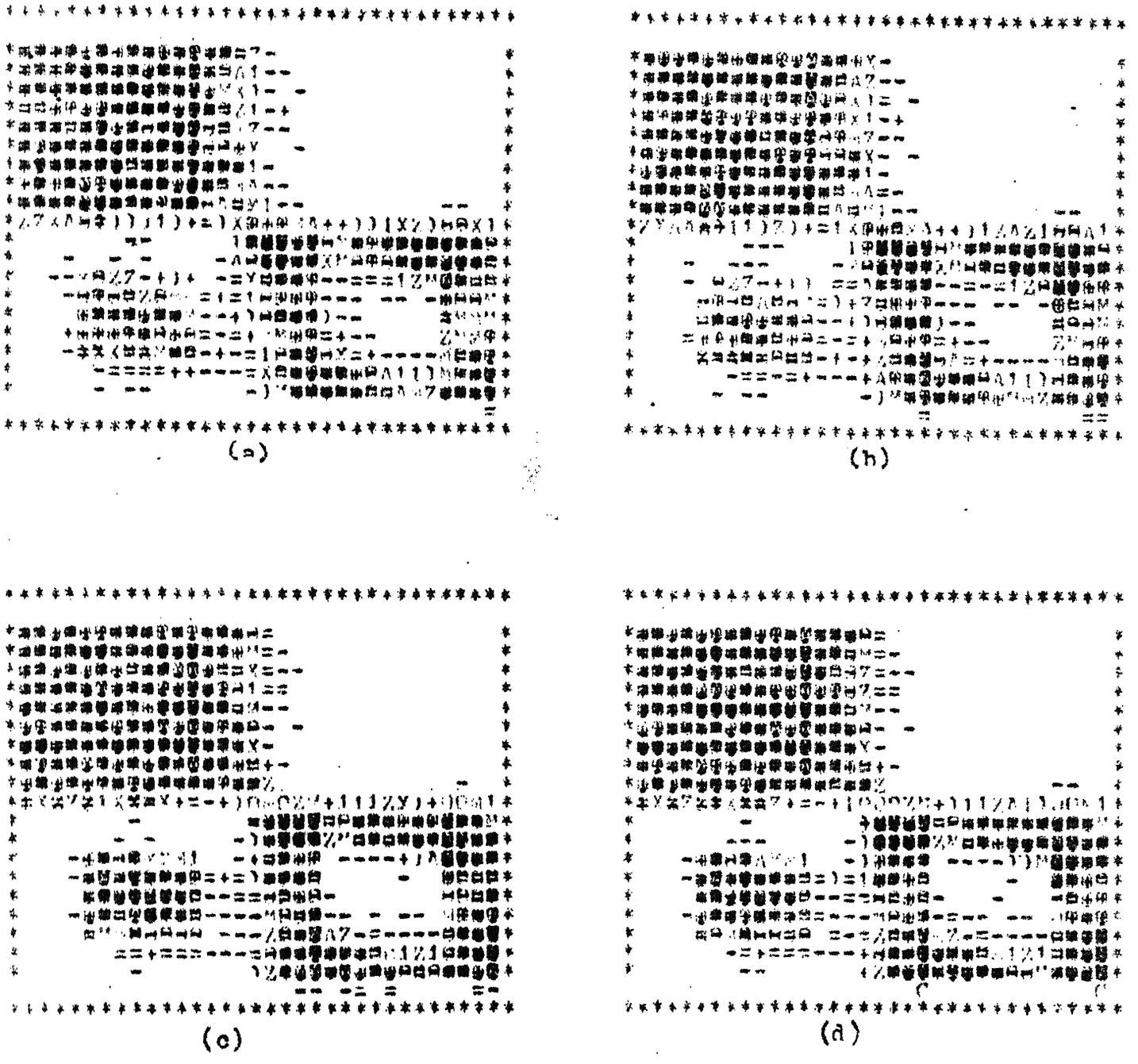


Fig.5.3 : Reconstruction of windows : NPR=10,NEQ=473,NIT=2, (a)PRA,(b)OPRA
(c) NPR=10, NEQ=473,NIT=10,PRA, (d)OPRA,NPR=10,NEQ=473,NIT=7.

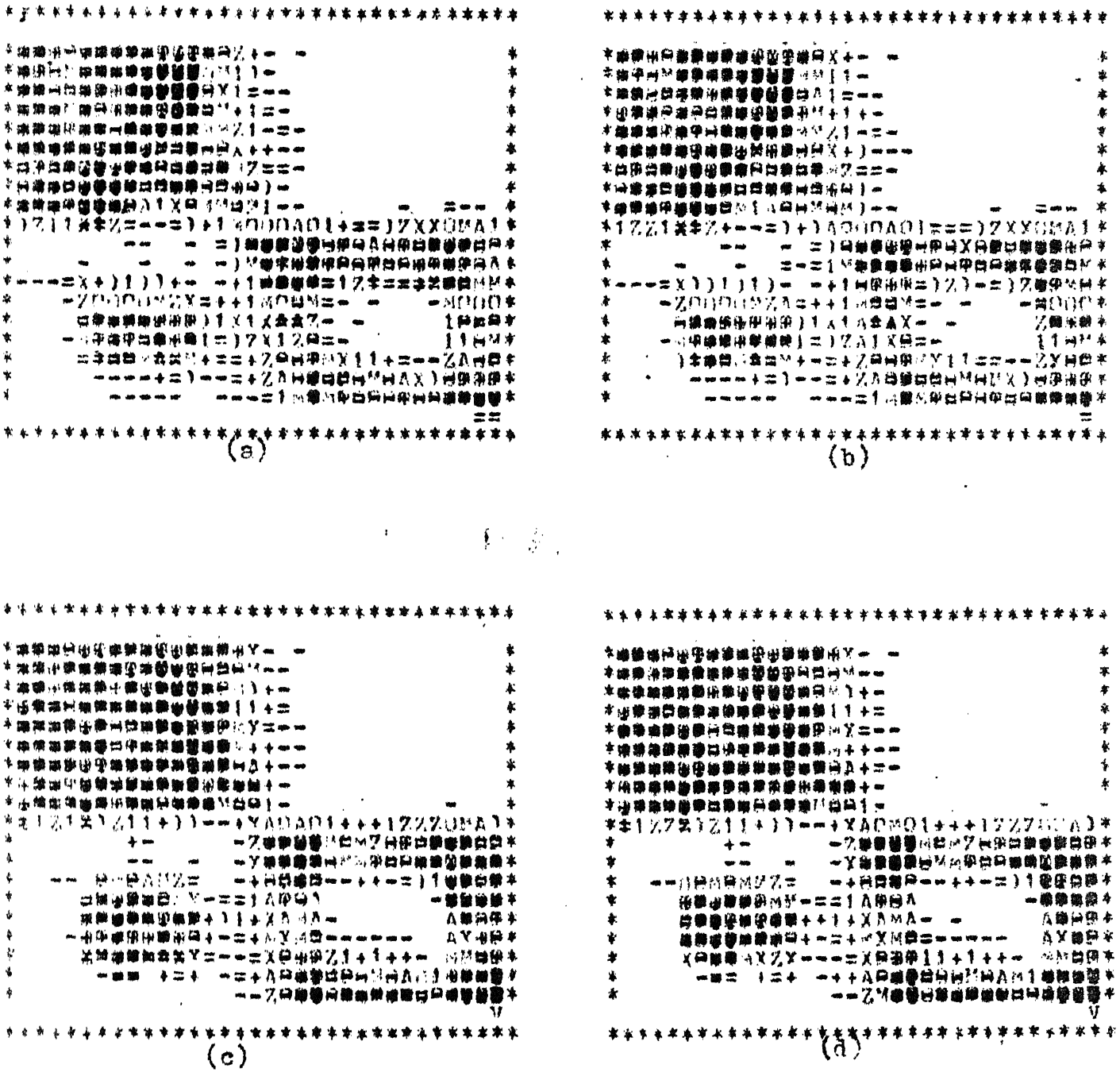
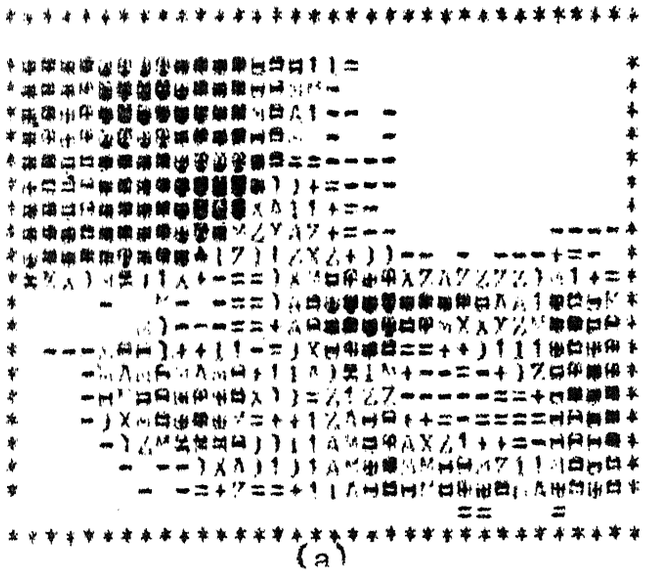
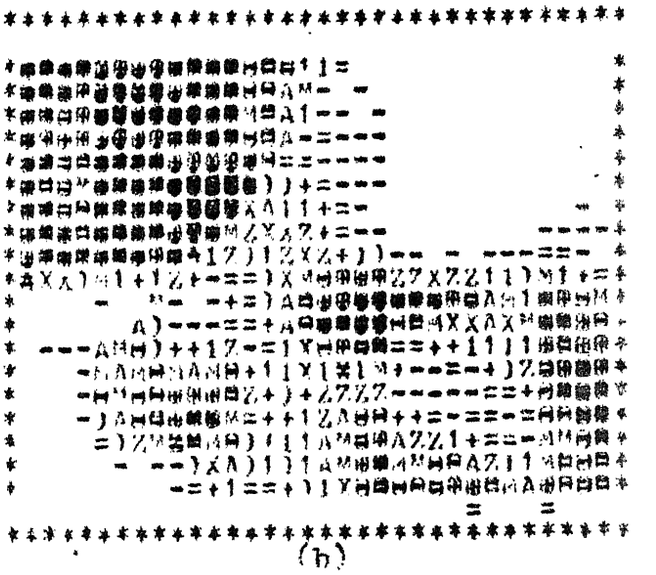


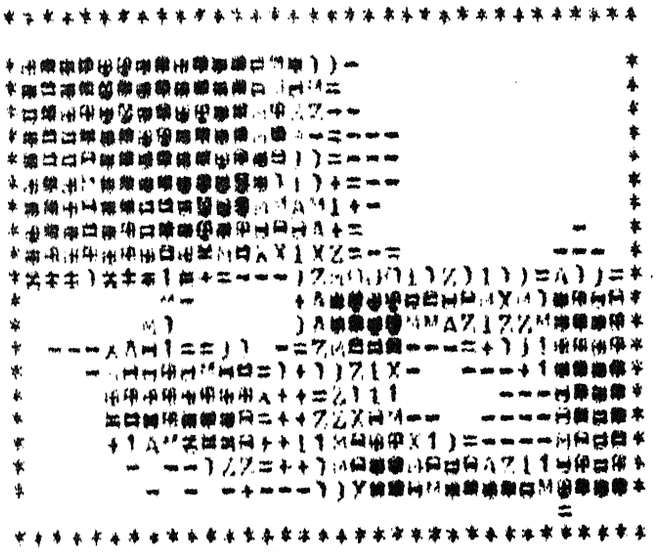
Fig.5.4 : Reconstruction of windows: NPR=8,NEQ=378,(a)PRA,NIT=2 (b) OPRA, NIT=2;(c)PRA,NPR=8,NEQ=378,NIT= ,(d)OPRA,NPR=8,NEQ=378,NIT=9.



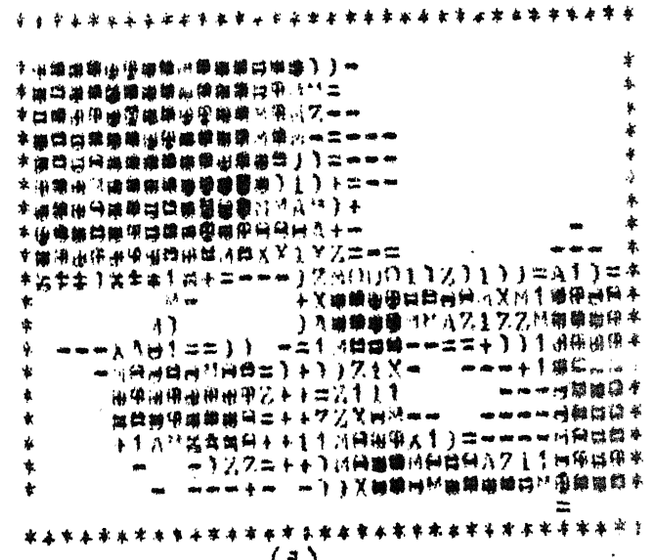
(a)



(b)

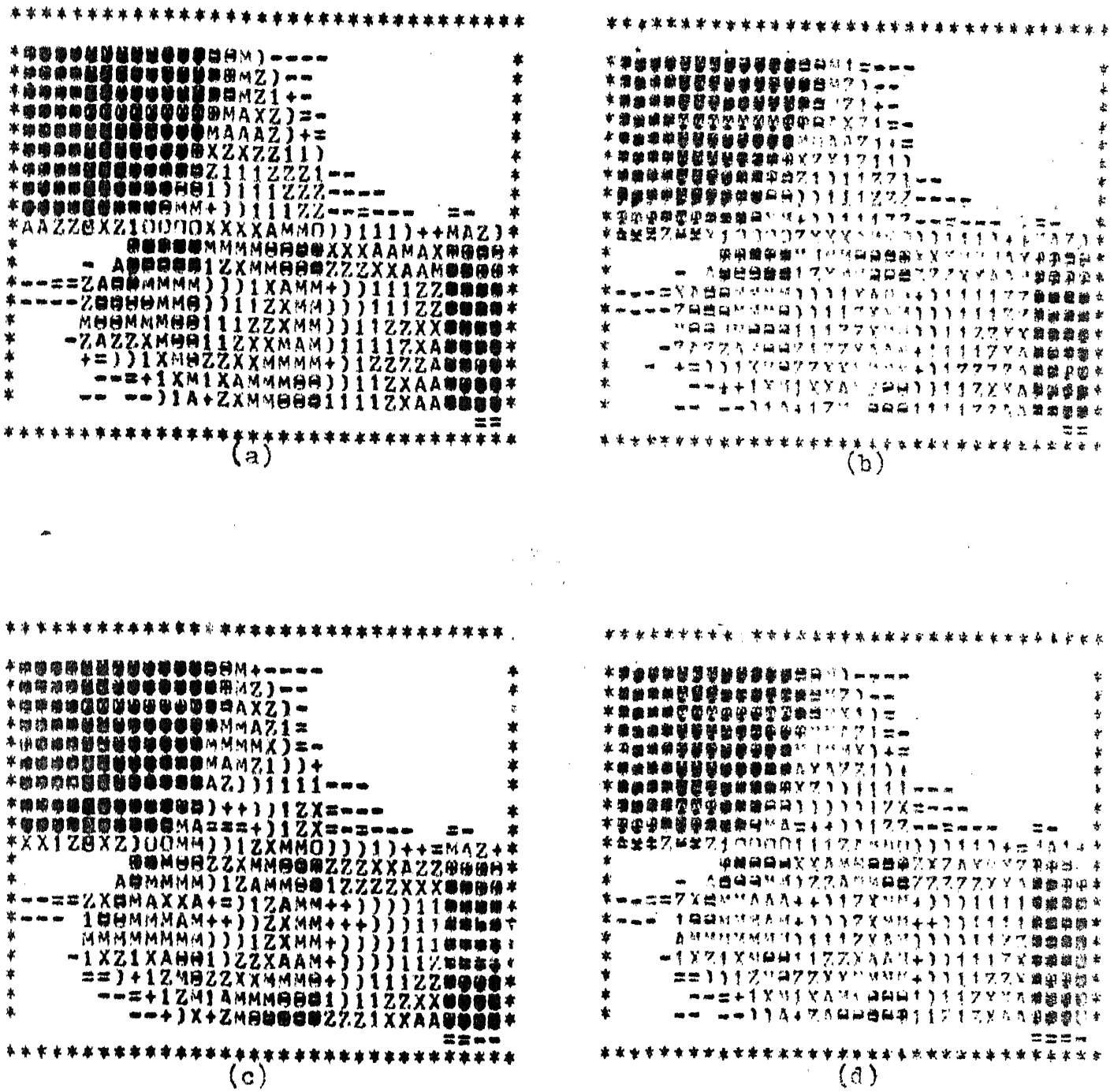


(c)



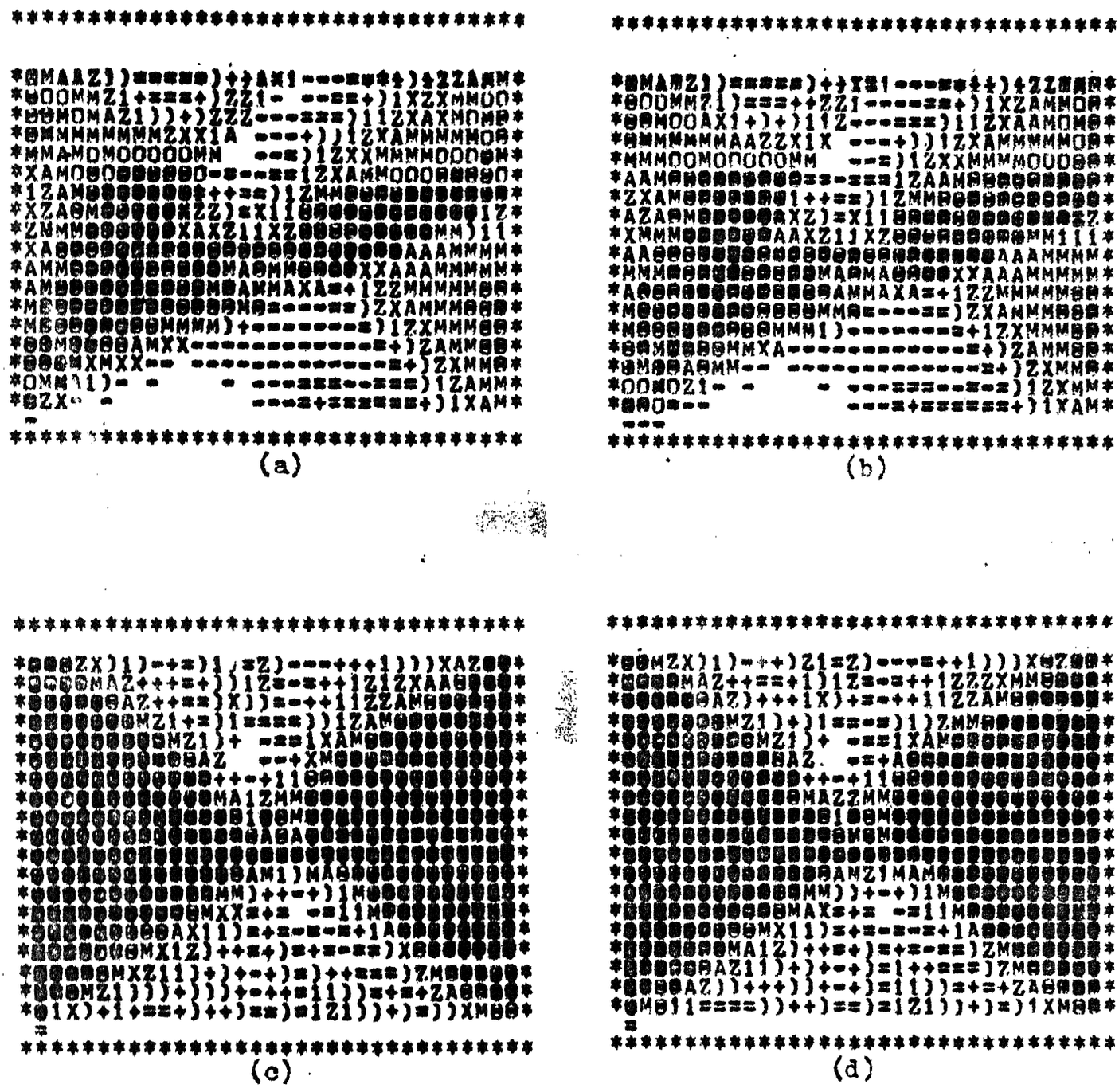
(d)

Fig.5.5 : Reconstruction of windows: NPR=6, NEQ=284, NIT=2; (a)PRA, (b)OPRA: NPR=6, NEQ=284, (c)PRA, NIT=11, (d)OPRA, NIT=10.



(a) (b) (c) (d)

Fig.5.6 : Reconstruction of windows : NPR=4, NEQ=189, NIT=2, (a) PRA, (b) OPRA; NPR=4, NEQ=189, (c) PRA, NIT=5, (d) OPRA, NIT=3.



(a) (b) (c) (d)

Fig.5.7 : Reconstruction of the pattern: NPR=6,NEQ=24,NIT=1,(a)PRA, (b) OPRA; NPR=6,NEQ=284,(c) PRA, NIT=14, (d) OPRA, NIT=11.

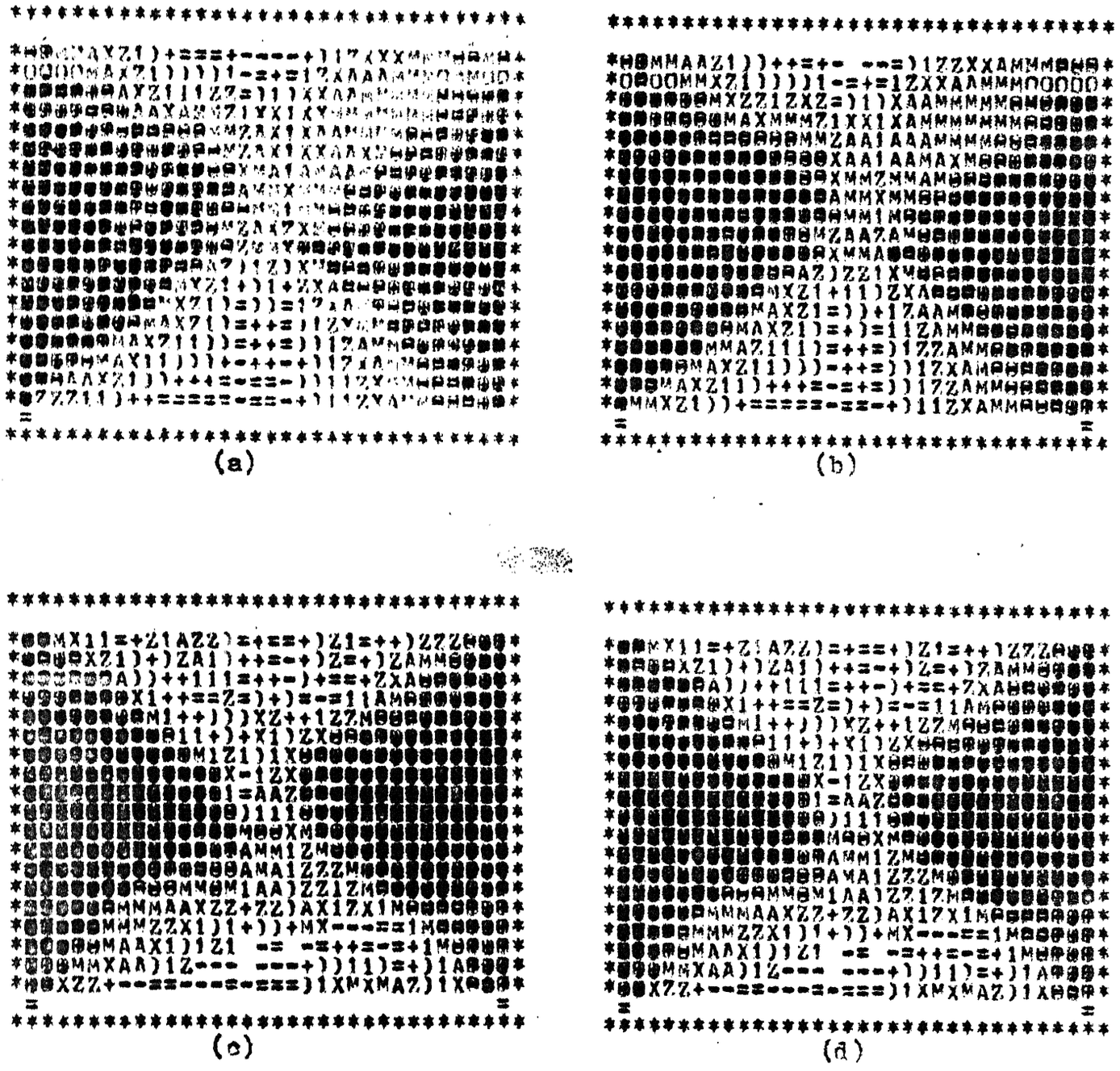
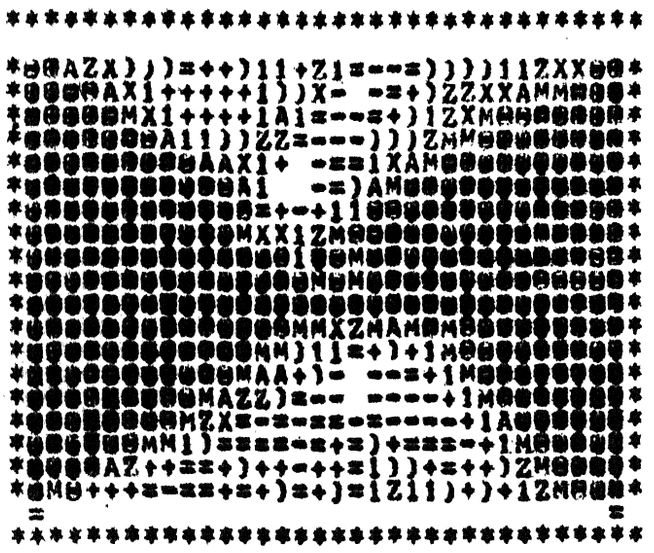
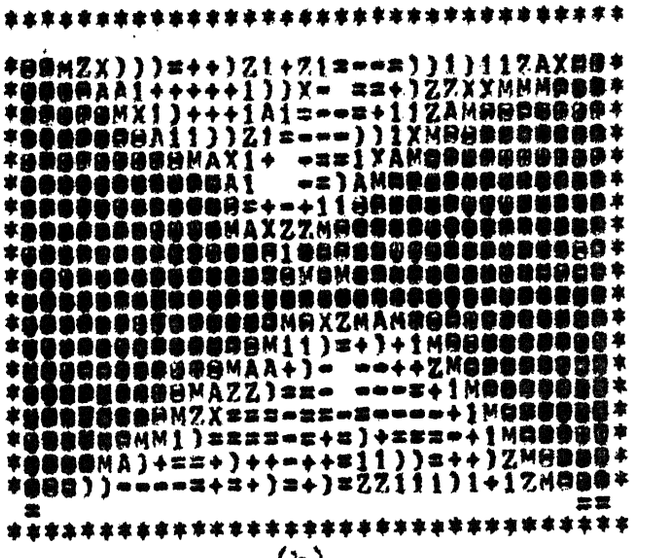


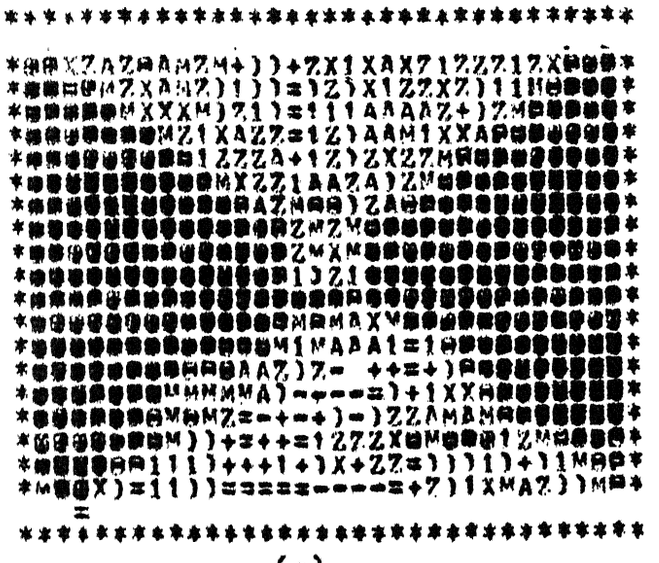
Fig. 5.8 : Reconstruction of the pattern: NPR=4, NEQ=189, NIT=3, (a) PRA, (b) OPRA; NPR=5, NEQ=232, NIT=3, (c) PRA, (d) OPRA.



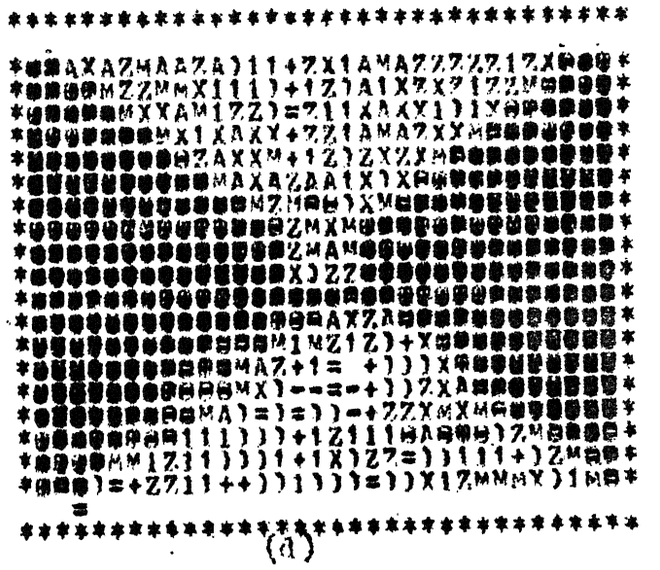
(a)



(b)

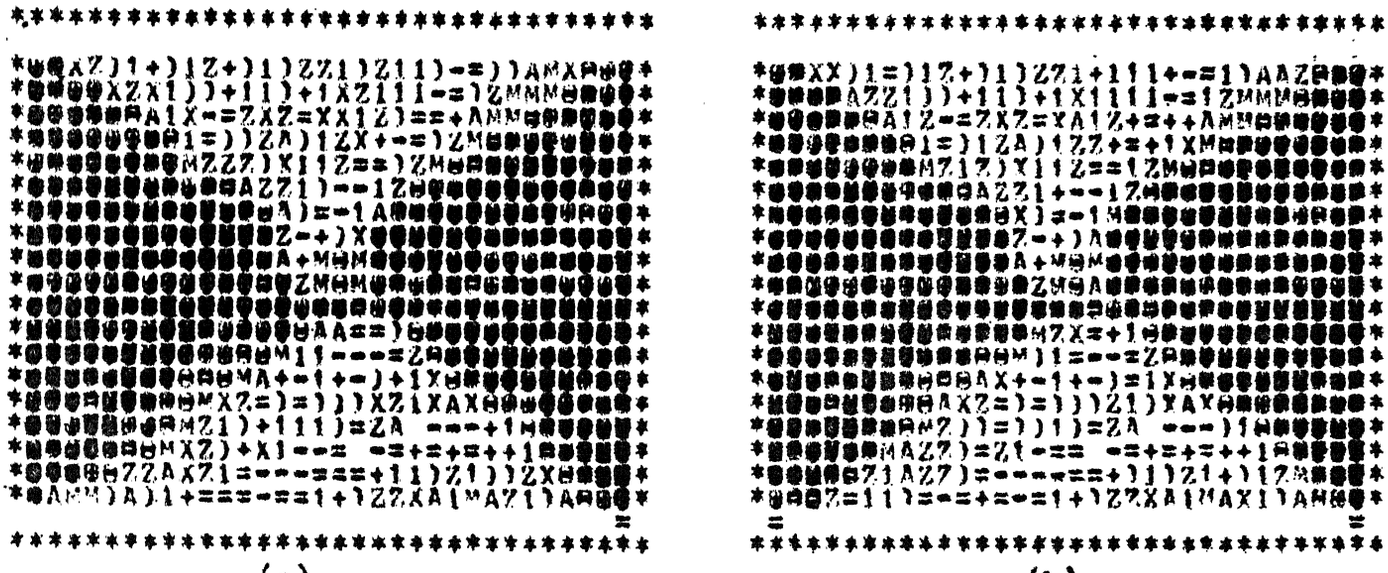


(c)

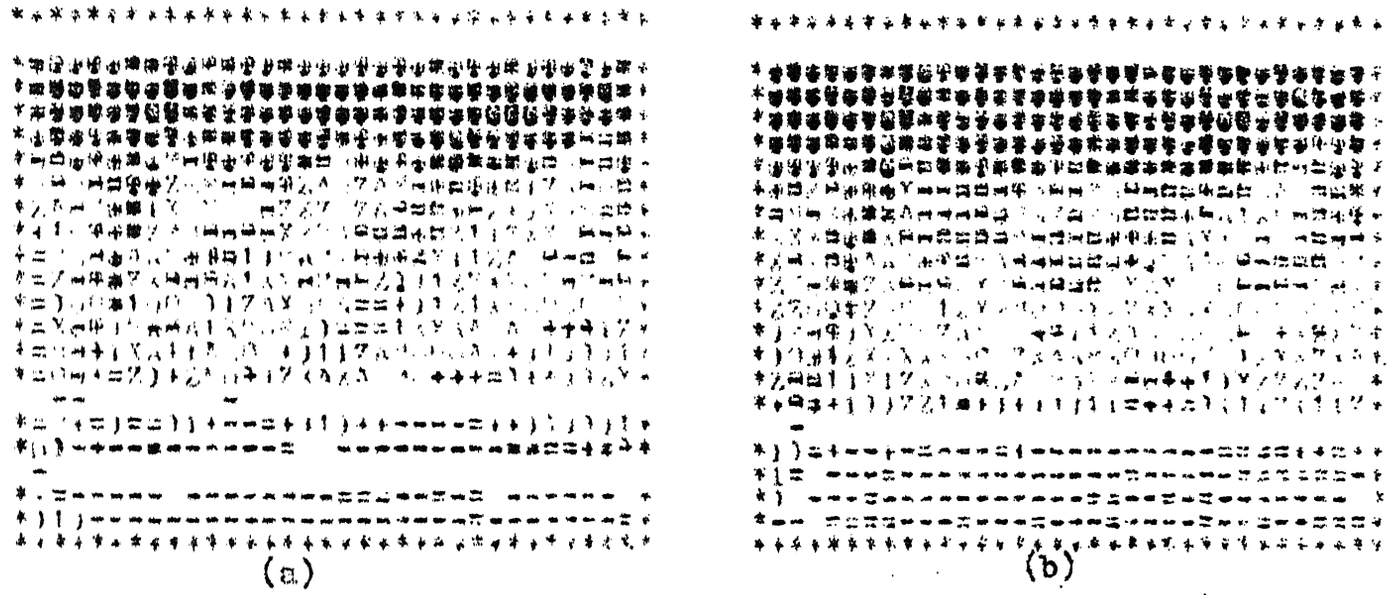


(d)

Fig. 5.9 : Reconstruction of the pattern : NPR=6,NEQ=284,NIT=3,(a) PRA, (b) OPRA; NPR=9,NEQ=424,NIT=3 (c) PRA, (d) OPRA.



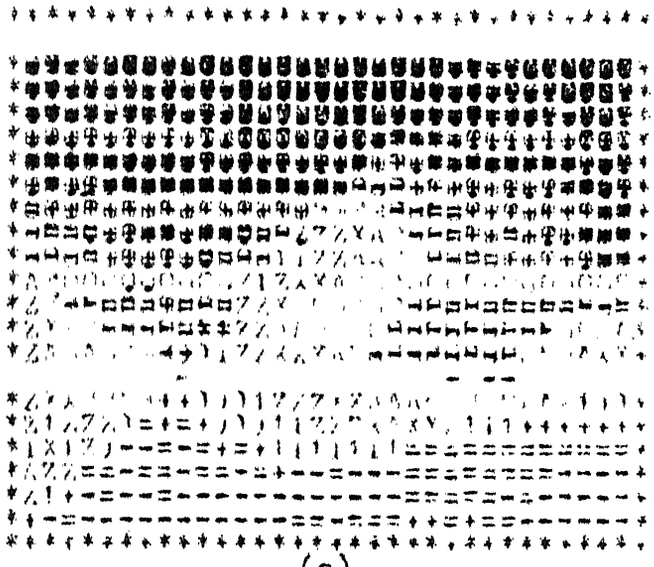
(a) (b)



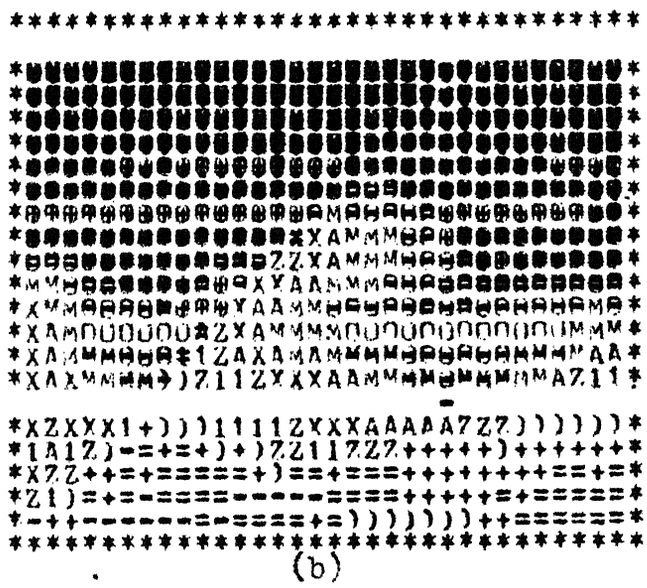
(a) (b)

Fig. 5.10(a) Reconstruction of the pattern: NPR=10, NEQ=473, NIT=3
(a) PRA, (b) OPRA

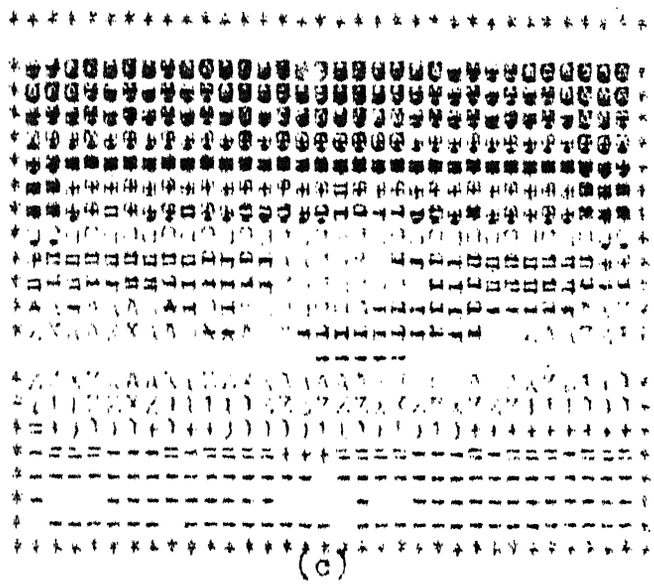
Fig. 5.10(b) Reconstruction of the graded square : NPR=11, NEQ=517, NIT=2
(a) PRA, (b) OPRA.



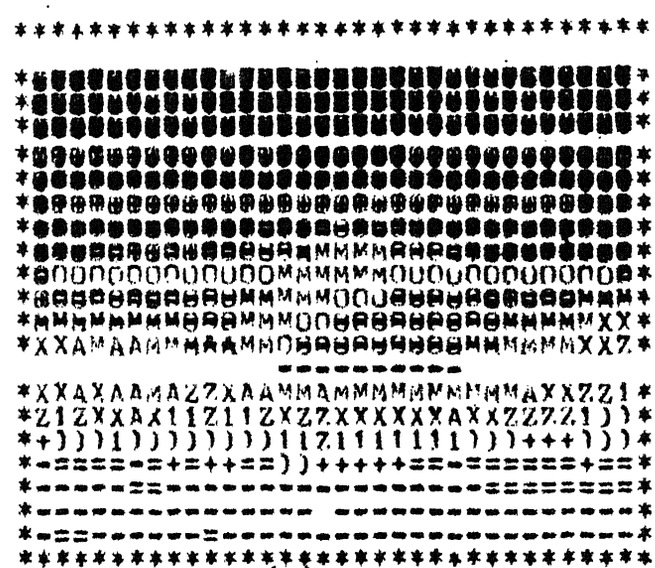
(a)



(b)

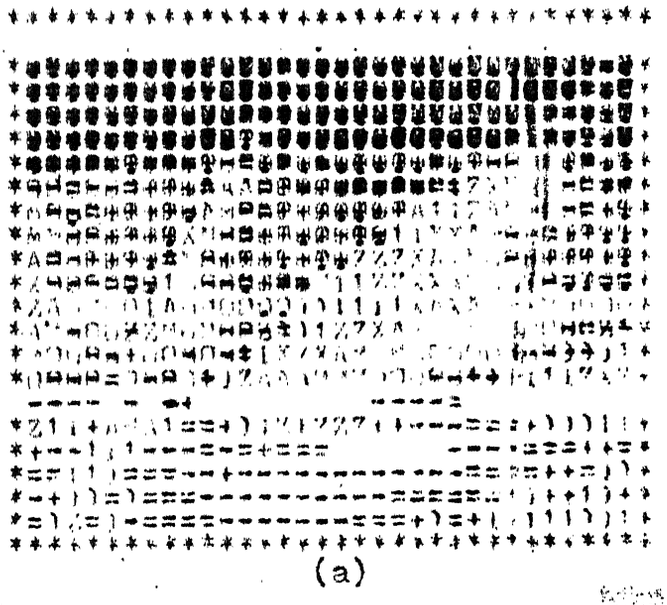


(c)

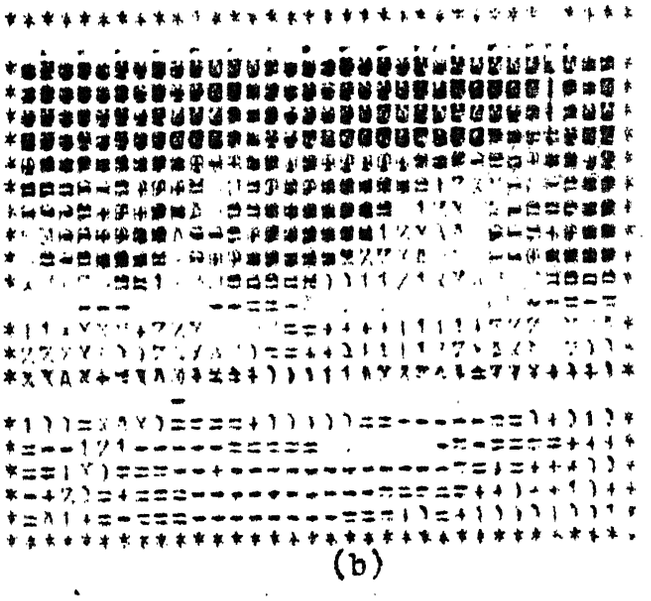


(d)

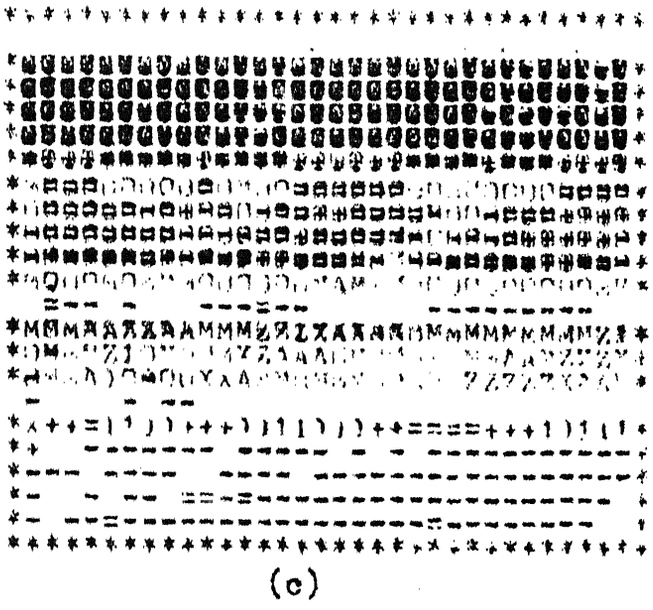
Fig. 5.11 : Reconstruction of the graded square : NPR=5, NEQ=232, NIT=2, (a) PRA, (b) OPRA; NPR=5, NEQ=232, (c) PRA, NIT=12, (d) OPRA, NIT=11.



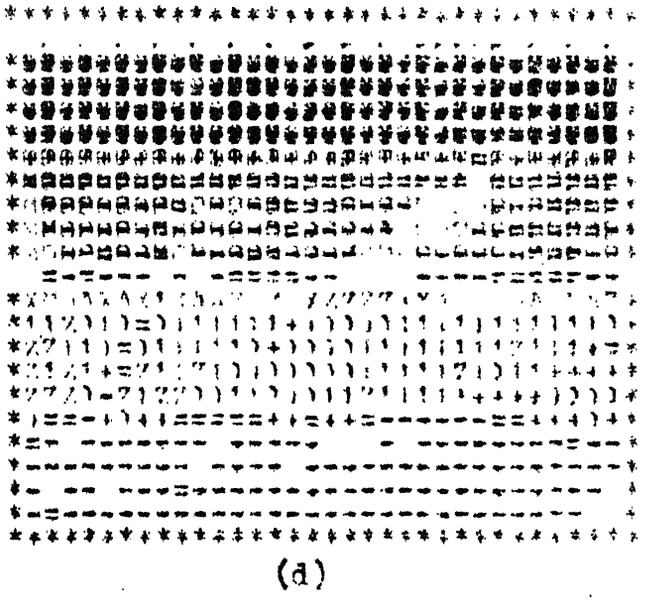
(a)



(b)



(c)



(d)

Fig. 5.12 : Reconstruction of the graded square : NPR=8, NEQ=378, NIT=2, (a) PRA, (b) OPRA; NPR=8, NEQ=378, (c) PRA, NIT=10, (d) OPRA, NIT=8.

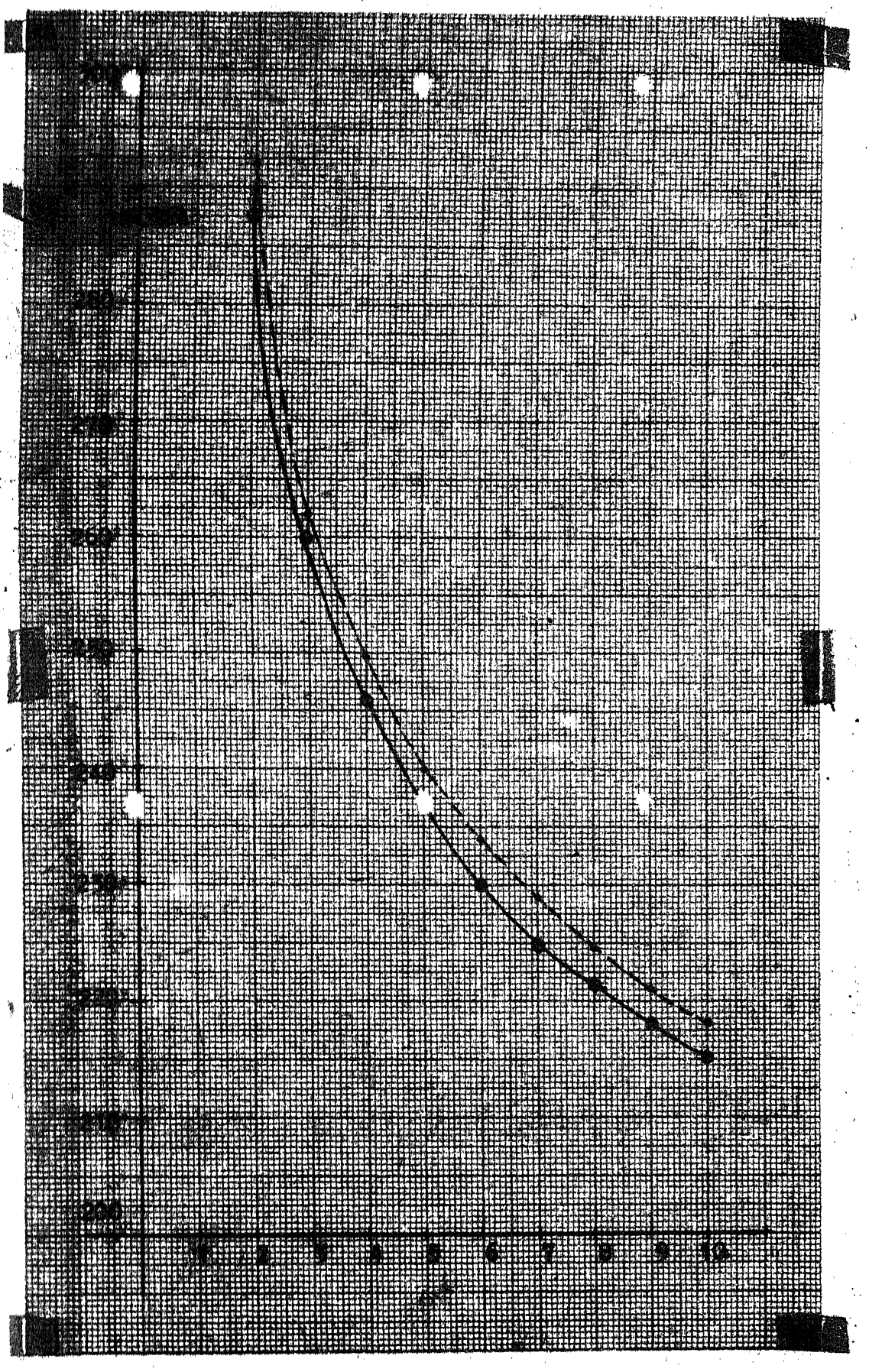
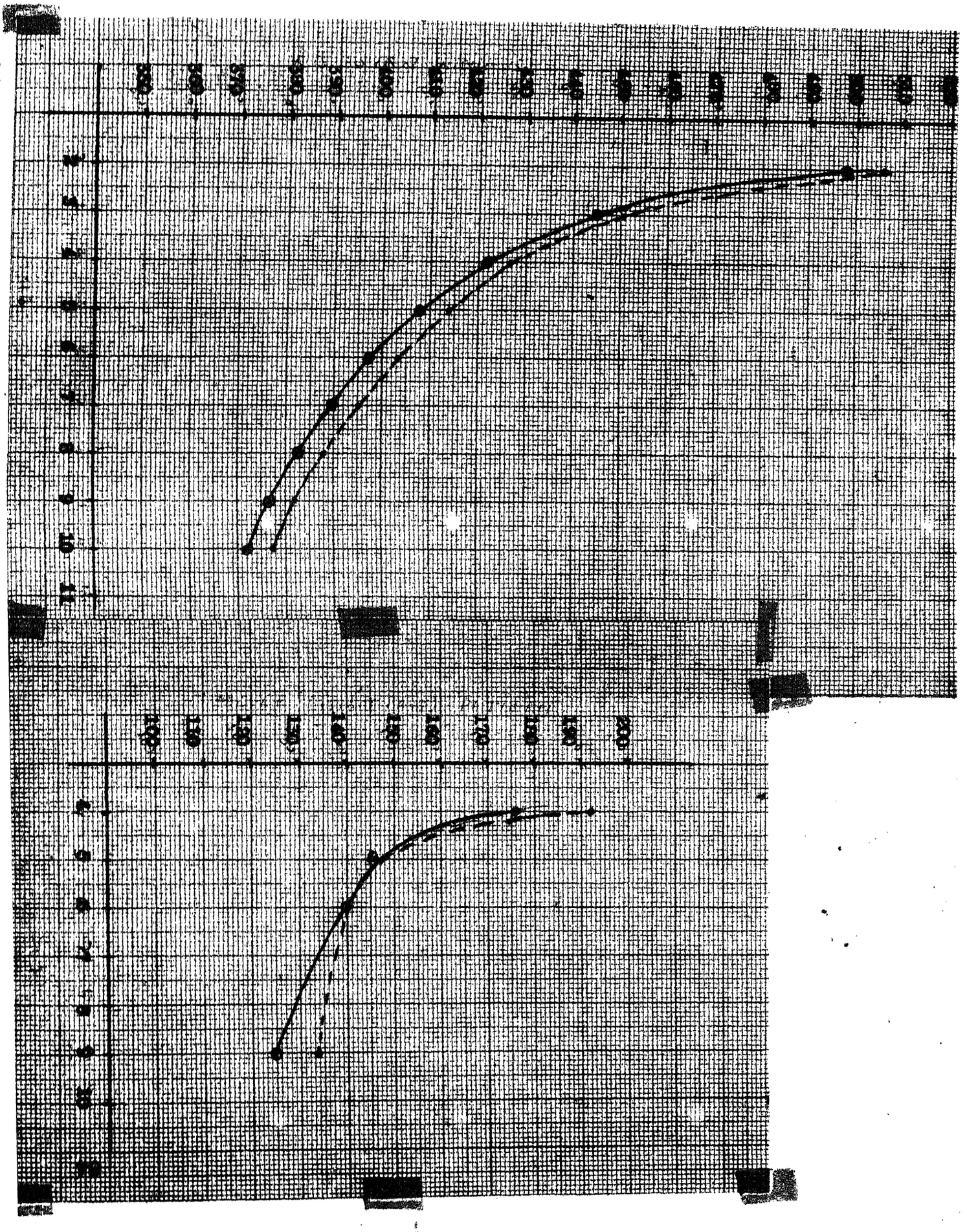


Fig. 5.15 MRERR, NPR = 10 (Windows)



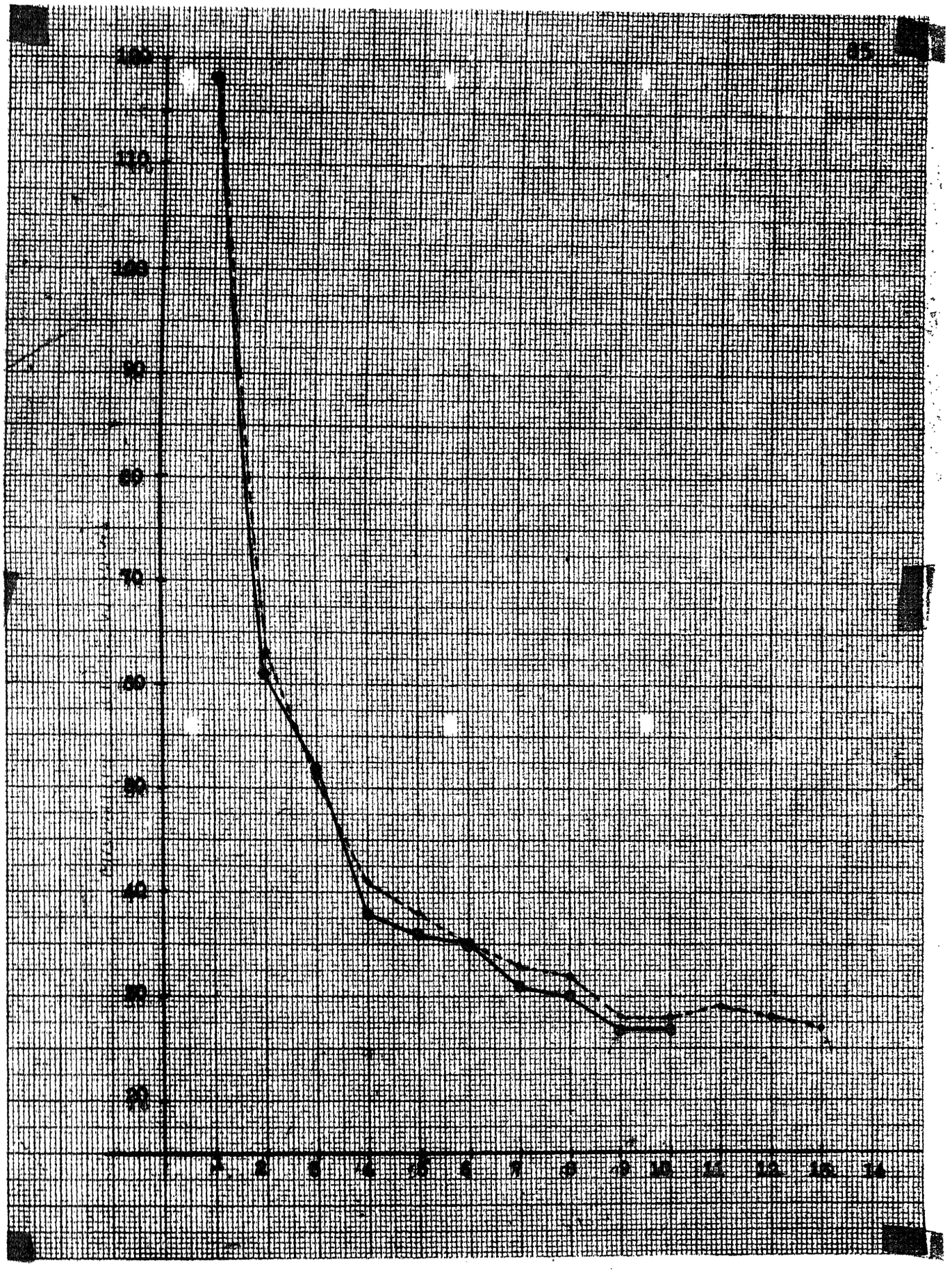


Fig. 5.17 Mismatches, NPR = 8 (Windows)

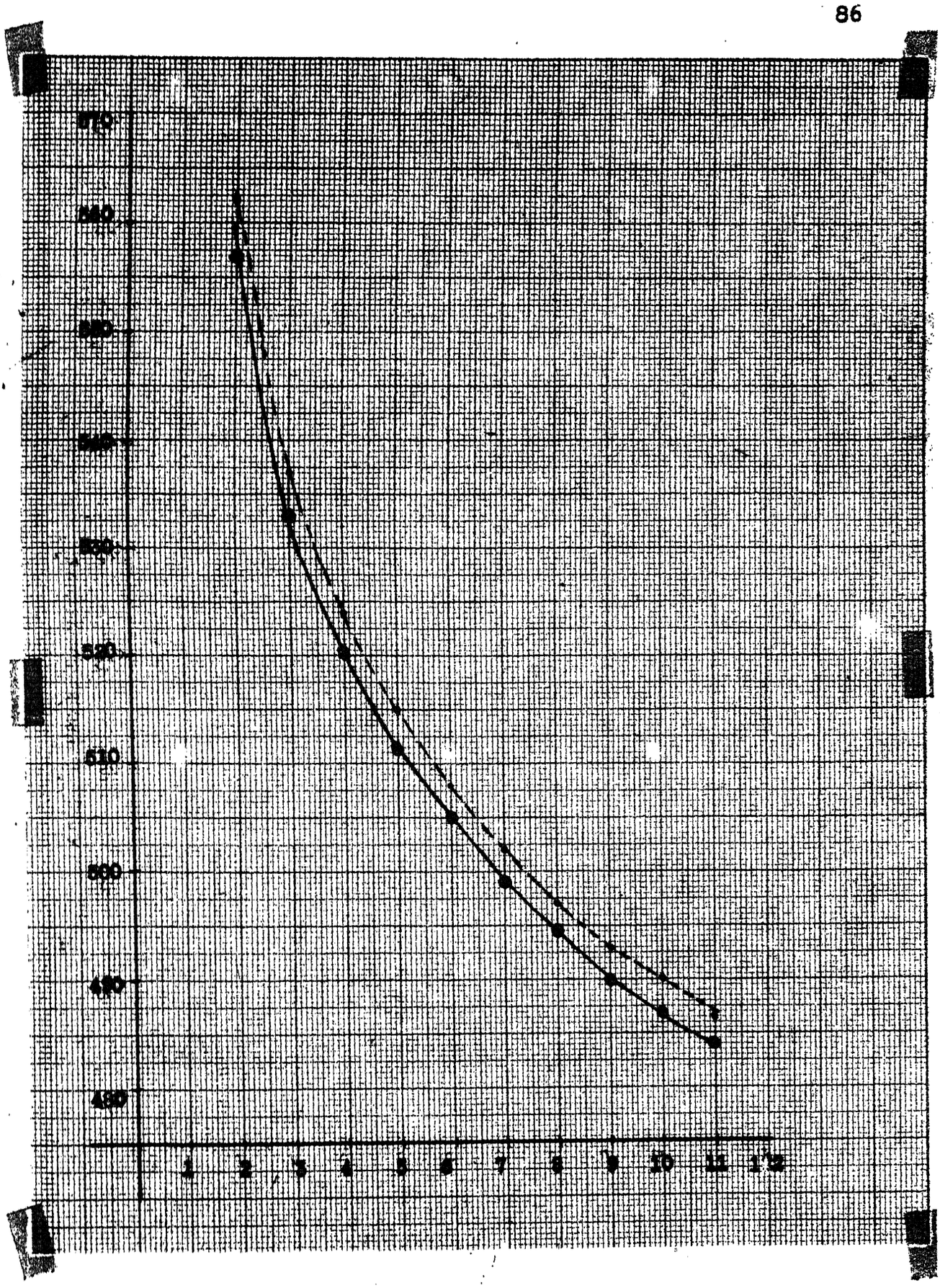


Fig. 5.18 RMS, NPR = 6 (Windows)

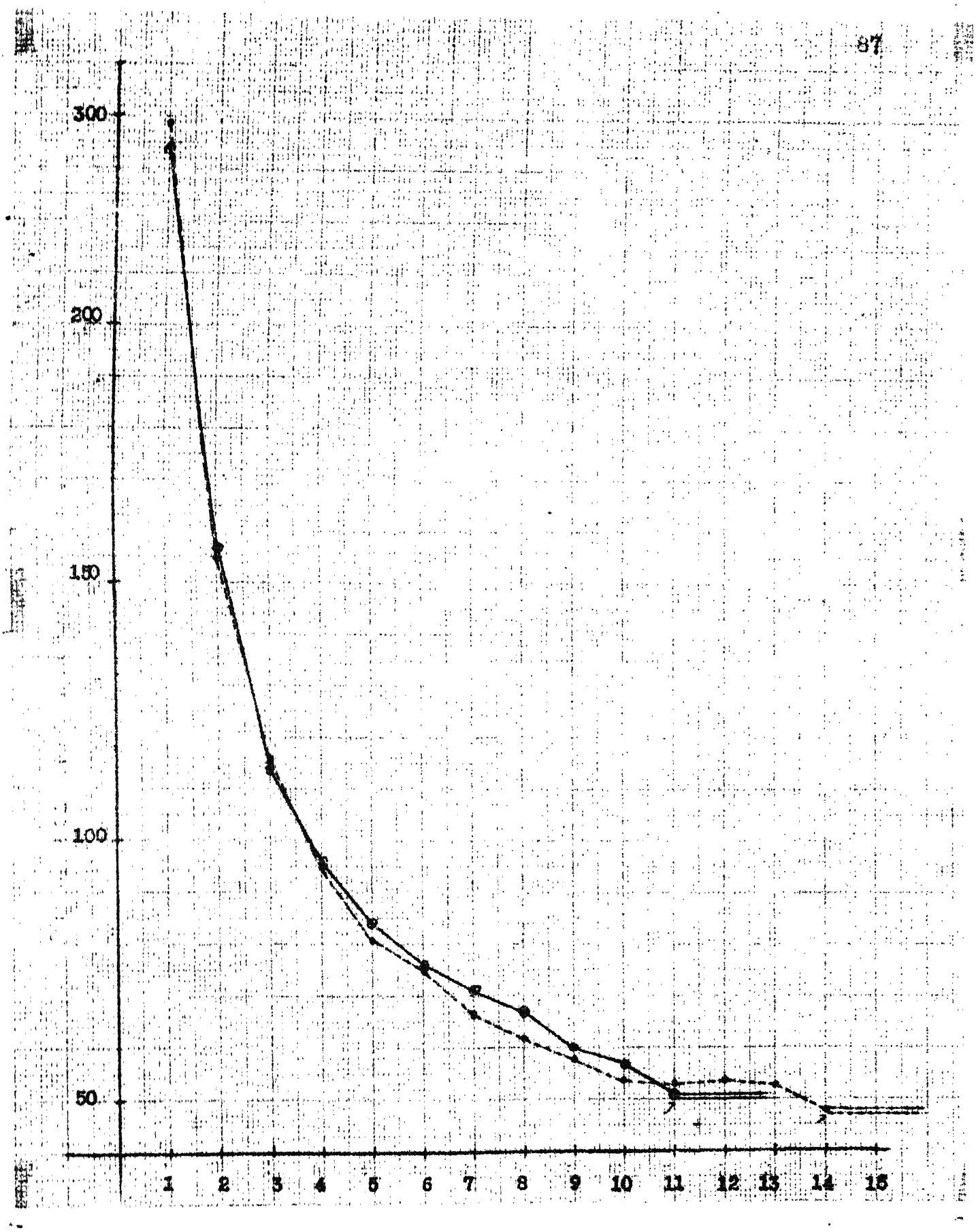


Fig. 5.19 Mismatches, NPR = 6 (pattern)

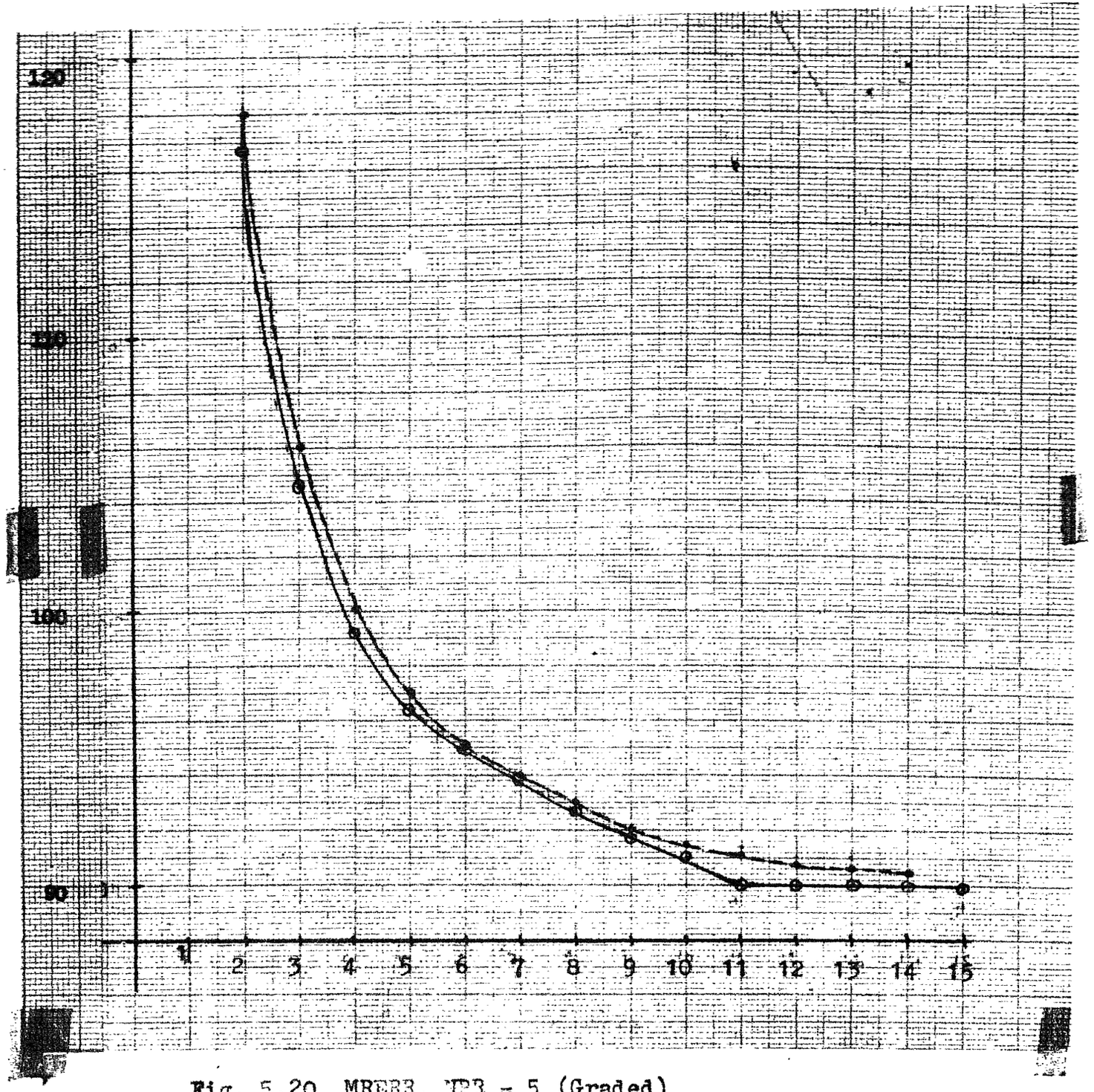


Fig. 5.20 MRERR, MPR = 5 (Graded)

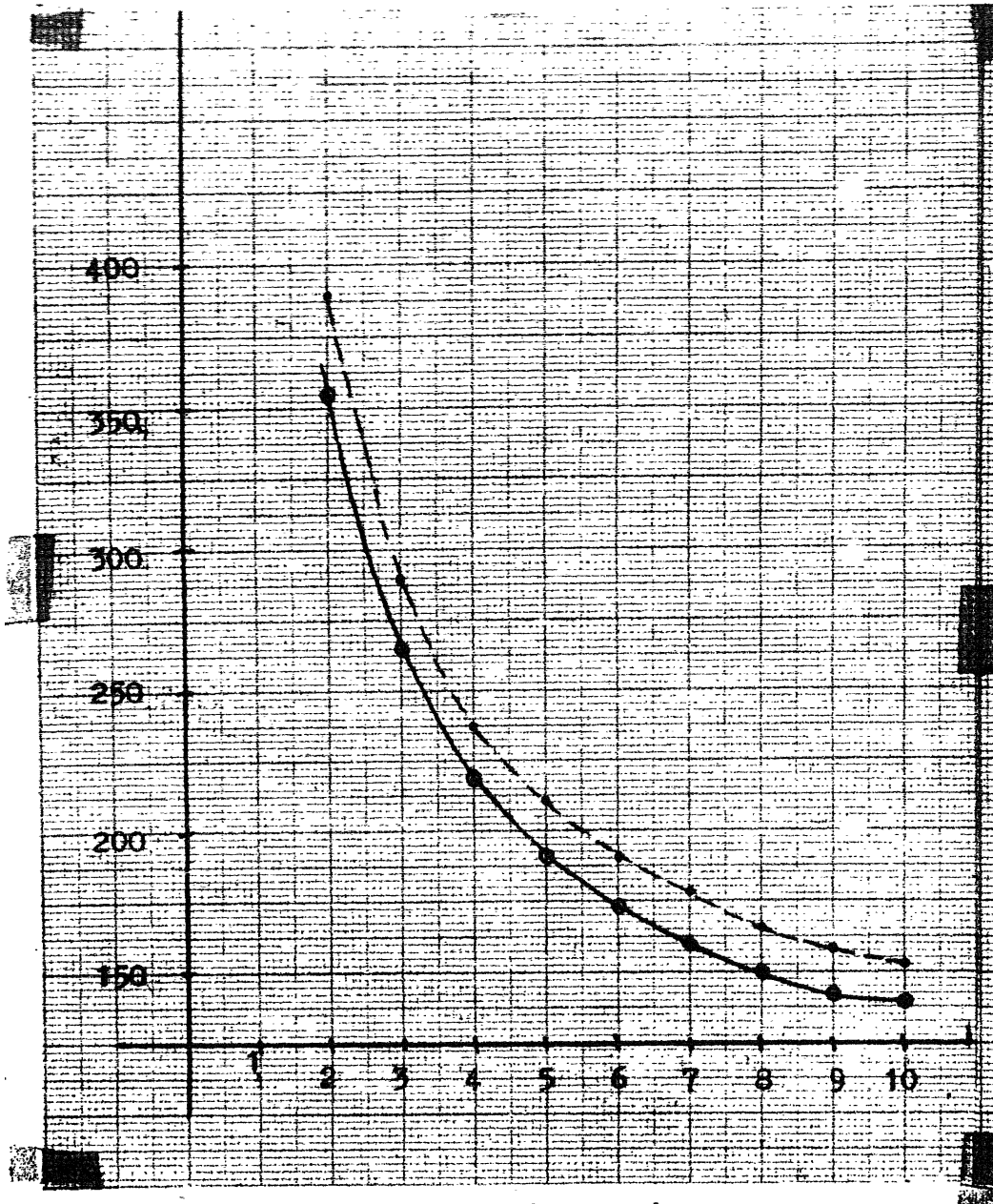


Fig. 5.21 RMS, NPR = 8 (Graded)

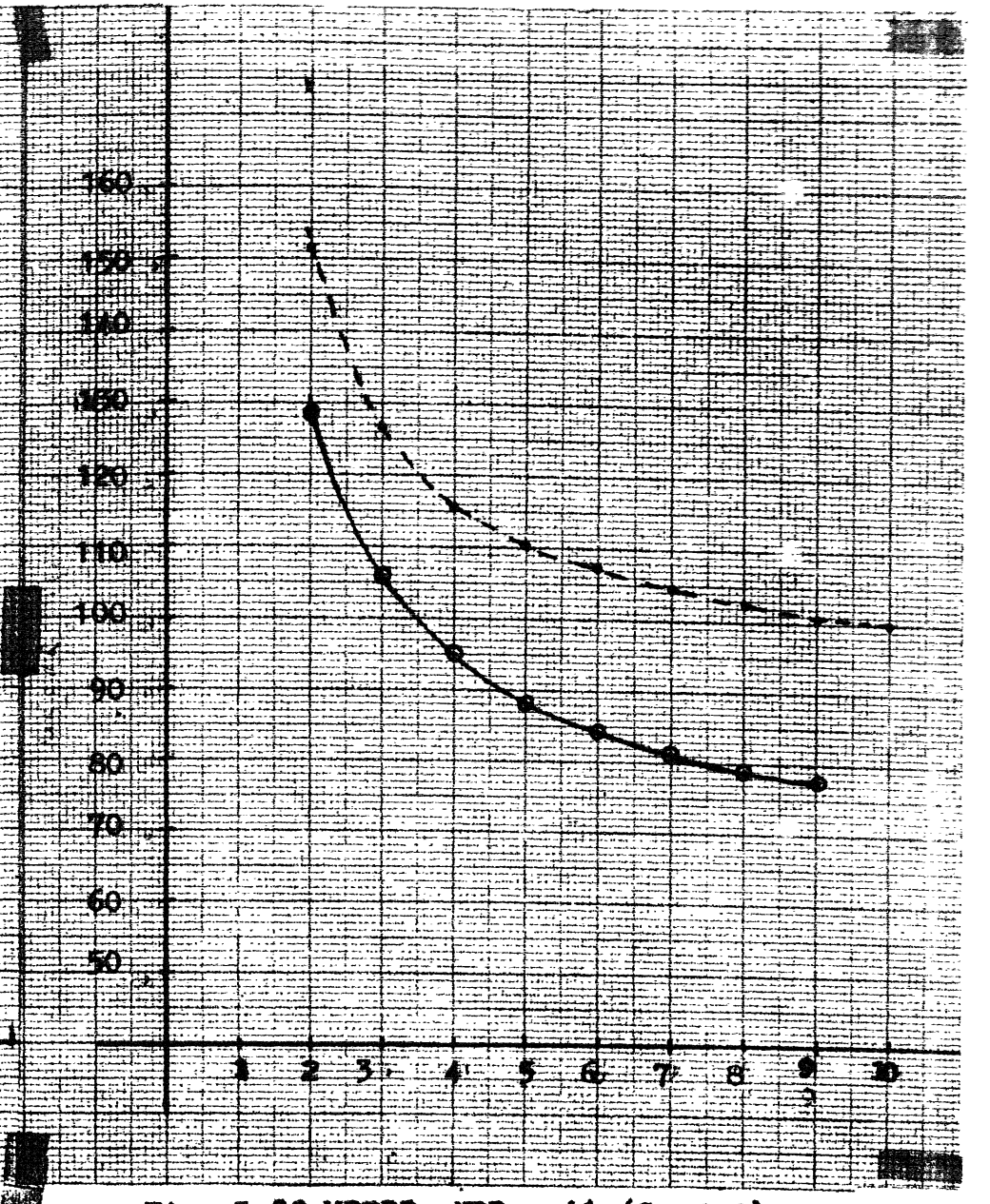


Fig. 5.22 MRERR, NPR = 11 (Graded)

REFERENCES

1. Lyons, E.C. Jr., Digital Image Processing- An Overview, Comtal Corporation, 1977.

2. Pratt, W., Digital Image Processing, John Wiley& sons, 1978.

3. Buddinger, T.F., Gullberg, G.T., Lawrence Berkeley Lab. Report, LBC-2146, 1974.

4. Andrews, J.R., American J. Roentgenol, 36, pp 575-587, 1936.

5. Gordon, R., Herman, G.T., Bender, R., Art for 3-D electron microscopy and X-ray photography, Journal theoret biology, 29, pp 471-481, 1970.

6. Gilbert, P.F.C., Iterative methods for reconstruction of 3-D objects from projections, J. theoret. Biology, 36, pp 105-117, 1972.

7. Kuhl, D.E., Edwards, R.Q., Ricci, A.R. and Reivich, M., Journal Nuc. Med., 14(4), pp 196-200, 1973.

8. Goitein, M., Nucl. Inst. Methods, 101(3), pp 509, 518, 1971.

9. Bracewell, R.N., Riddle, A.C., Inversion of Fan beam scans in radioastronomy, Astrophysics, J., 150, pp 427-434, 1967.

10. Schmidlin, P., Proc. Third International Conf. on Data Handling and Image Processing in Scintigraphy, Boston, 1973.

11. DeRosier, D.J., Klug, A., Reconstruction of three-dimensional structures from electron micrographs, Nature, 217, pp 130-134, 1968.

15. Krishnan, S., Prabhu, S., Krishnamurthy, E.V., Probabilistic reinforcement algorithms for the reconstruction of pictures from their projections, Int. J. Syst. Sc., 4(4), pp 661-670, 1973.

16. Vainshtein, B.K., Finding the structure of objects from projections, Sov. Phys. Crystall., 15(5), pp 781-787, 1971.

17. Wernecke, S.J., and D'Addario, Larrym, D., Maximum entropy image reconstruction, IEEE Trans. on Compt. C-26(4), pp 351-364, 1977.

18. Radon, J., Uber die Bestimmung Von Funktion durch ihre Integral Werte langs gewisser Mannig faltig keiten (on the determination of functions from their integrals along certain manifolds), Math. Phy. Klass 69, pp 262-277, 1917.

19. Ramachandran, G.N., Reconstruction of substance from shadow-I, Proc. Ind. Aca. Sc., 74, pp 14-24, 1971.

20. Ramachandran, G.N., and Lakshminarayanan, A.V., Three dimensional reconstruction from radiographs: application of convolution methods instead of fourier transforms, Proc. Nat. Acad. Sc. U.S., 68(9), pp 2236-2240, 1971.

21. Gordon, R., A tutorial on ART, IEEE trans. on Nuc. Sc., NS 24(3), pp 78-93, 1974.

22. Radon, J., Ber. Sacchs. Akad. Wiss Leipzig, matr., Physiki, 69, 262-277, 1917.

23. Gordon, R., Herman, G.T., Reconstruction of pictures from their projections, Comm. of the ACM, 14(12), 1971.

24. Bender, R., Bellman, S.H., Gordon, R., ART and the Ribosome: A prelim report on the 3-dimension structure of individual ribosome determined by an ART, J. theoretical biol. 29, pp 403-487, 1971.

25. Wee, W.G., and Hsieh, Tsing Tao, An application of projection transform in image transmission, IEEE trans. sys. man and cyb. SMC-6 (7), pp 486-493, 1976.

26. Oppenheim, B.E., More accurate algorithms for iterative 3-D reconstruction, IEEE trans. on Nuc. Sc., NS-21, pp 72-77, 1974.

27. Cho, Z.H., Advances in picture reconstruction: theory and applications, Compt. in Biol. & Med., 61(4), 1974.

28. Rosenfeld, A., Picture processing by computer, Computer Survey, pp. 146-176, 1969.

29. Singh, S., Reconstruction from projections - a comparative study, M.Tech. thesis (IIT Kanpur, India).

30. Kaczmarz, S., An zenaherte Auflosung von systemen Linearer Gleichungen, Bull. Acad. Polon. Sciences et Letters, A., pp. 355-357, 1937.

31. Tanabe, K., Projection method for solving a singular system of linear equations and its applications, Numerische Mathematik, 17, pp. 203-214, 1971.

32. Simmons, G.F., Introduction to Topology and Modern Analysis, McGraw-Hill Kogakusha.

33. Ramakrishna, R.S., Some iterative techniques in digital image restoration, Ph.D. Thesis (IIT Kanpur, India).

34. Frank, L.E., Signal theory, Prentice Hall Engle Wood Cliffs, New Jersey, 1969.

35. Papoulis, A., Signal Analysis, McGraw Hill Company, New York, 1977.

36. Yosida, K Osaku, Functional Analysis,

37. Bellman, S.H., Bender, R., Gordon, R., and Rowe, J.E.,Jr., ART is science, being a defence of algebraic reconstruction techniques for three dimensional electron microscopy, J. theor. Biol., 32, 203-216, 1971.

38. Herman, G.T., Two direct methods for reconstructing pictures from their projections : a comparative study, Comput. Graphics Image Process, 1(2), pp. 123-144, 1972.

39. Herman, G.T., and Rowland, S., Three methods for reconstructing objects from X-ray: A comparative study, Comput. Graphics Image Processing, 2, pp. 151-178, 1973.

40. Schmidlin, P., Iterative seperation of section in tomographic scintigrams, Nuc, med. (Stuttgart), 1, pp.1-16, 1972.

41. Gordon, R., Herman, G.T., Three dimension reconstruction from projections: A review of algorithms, Proc. of the Society of photo-optical Instr. Engineers, 47, 1975.

42. Kashyap, R.L., Mittal, M.C., in Proc. First Int. joint Conf. Pattern Recognition, Washington, pp 286-292, 1973.

43. Dholme, P.E., Acta Radiol. Suppl., 193, pp.1-109, 1960.

44. Schmidlin, P., Medical radioisotope scintigraphy 1972 (International Atomic Energy Agency, Vienna, 1973).

45. Lee, D.E., Keyes, J.W., Simon, N., in application of optical instrument in medicine II, Proc. Soc. Photo optical Instr. Engg., 1973.

46. Keyes, J.W. Jr., and Simon, W., Proc. Third Symp. on Sharing of computer programs and technol. in Nucl. Medicine, Miami, Florida, USAEC, Report Conf. 730627, pp. 190-201, 1973.

47. Peters, T.M., Ph.D. Thesis, University of Canterbury, Newzealand, 1973.

48. John, F., Plane waves and spherical mean applied to partial differential equations (Interscience, New York, 1955).

49. Berry, M.V., and Gibbs, D.F., Proc. Roy. Soc. A314, pp. 143-152, 1970.

50. Cormack, A.M., Phys. Med. Biol., 18(2), pp. 195-207, 1973.

51. Rosenfeld, A., Compactness of figures in digital pictures, IEEE Trans. on Sys. man & cyb., 1974.

52. Rosenfeld, A., Picture processing by computer, NY, Academy Press, pp 161-162, 1969.

```
      C     ********************************************************************
      C     PROGRAM FOR CONSTRUCTING AND RECONSTRUCTING WINDOW PATTERN
      C     USING THE PROJECTION METHOD WITHOUT ORTHOGONALIZATION.
      C     ********************************************************************
            DIMENSION B(1024),F(1024)
            N=32
            DO 10I=1,N
            DO 10J=1,N
            IJ=(I-1)*N+J
   10       B(IJ)=0.
            DO 20I=1,N
            DO 30J=1,N
            IJ=(I-1)*N+J
            IF((I.GT.4.AND.I.LE.12).AND.(J.GT.4.AND.J.LE.12))B(IJ)=1.0
            IF(J.GT.16.AND.I.LE.16)B(IJ)=1.0
            IF(I.GT.16.AND.J.LE.16)B(IJ)=1.0
            IF((I.GT.20.AND.I.LE.28).AND.(J.GT.4.AND.J.LE.12))B(IJ)=0.
   30       CONTINUE
   20       CONTINUE
            CALL PRF(B,N)
            STOP
            END
```

```
C
C     RECONSTRUCTION OF THE FIGURE OF THE ELEPHANT USING PRF.
C
      DIMENSION A(1024),F(1024),K(200),MF(1024)
      INTEGER B,P,PP,S
      REAL G(20),C(1024)
      DATA A/1024*0/
      DATA F/1024*0/
      DATA G/20*0/
      CALL ANIMAL(C,N)
      N=32
      PI=3.1429570
      PP=4
      DO 5L6=1,10
      MARK=0
      TYPE*,MARK
      IF(MARK.LT.2)GO TO 10
      B(1)=0;B(2)=0;B(3)=0;B(4)=0;B(5)=0;B(6)=0;B(7)=0;B(8)=0
      B(9)=0;B(10)=0;S=0
      THETA=0.;S=1
10    IF(G(S).LT..5)GO TO 20
      IF(G(S).GT.1.5)GO TO 80
      GO TO 70
20    A(S)=A(S)+.5
      CALL PROJ(THETA,C,B,P,A,P,K,P,S)
      IF(P.LE.1)GO TO 20
      IF(MARK.EQ.1)MF0=MF0+1
      PP=2*I
      CALL PROD(A,F,Y)
      CALL PROD(A,A,Z)
      IF(Z.EQ.0.)GO TO 10
      DO 1001=1,PP
      F(K(I))=F(K(I))-(Y-P)/Z*A(K(I))
      IF(F(A(I)).LT.0.)F(A(I))=0
100   THETA=THETA+PI/4.;S=S+1
70    IF(S.GT.1)GO TO 100
      IF(G(S).GE.1.5)GO TO 70
      A(3)=A(S)+.5
80    DO 90 I=1,1024
90    A(I)=0.
      CALL PROJ(THETA,C,B,P,A,P,K,P,S)
      IF(P.GE.1)GO TO 70
      IF(MARK.EQ.1)MF0=MF0+1
      PP=2*I
      CALL PROD(A,F,Y)
      CALL PROD(A,A,Z)
      IF(Z.EQ.0.)GO TO 10
      DO 1001=1,PP
      F(K(I))=F(K(I))-(Y-P)/Z*A(K(I))
      IF(F(A(I)).LT.0.)F(A(I))=0.
11    LSS=0
160   DO 170 I=1,4
      IF(G(S).GE.1.5)LSS=LSS+1
170   IF(LSS.EQ.4)GO TO 180
      IF(S.LT.4)GO TO 70
      GO TO 10
      PRINT 350
      FORMAT(1H ,' RECONSTRUCTION WITHOUT ORTHOGONALISATION.')
350   PRINT 300,PP,MF0,MARK
180   FORMAT(1H ,'PP =',I4,'  MF0 =',I5,'  ITERATIONS =',I5)
300   FORMAT(1H ,32F5.2)
200   CALL PIC(*,0)
C     MIS=0
      DO 6I=1,1024
      MF(I)=0
      IF(F(I).GE..5)MF(I)=1
6     DO 7I=1,1024
      IF(C(I).NE.MF(I))MIS=MIS+1
7     PRINT 222,MIS
      FORMAT(1H ,'NO. OF MISMATCHES = ',I5)
222   CALL PIC(*,0)
      CALL PRO(C,F,N)
      PRINT 244
      FORMAT(1H ,///)
244   STOP
      END
```

```
00090 C
00095 CCCCCC
00100 CCCCCC  GENERATION AND RECONSTRUCTION OF A 3-DENSITY PATTERN WITHOUT
00110 CCCCCC  ORTHOGONALIZATION.
00120 CCCCCC
00130 CCCCCC
00140 C
00200       DIMENSION B(1024)
00300       N=32
00400       DO 10I=1,1024
00500 10    B(I)=0.45
00600       II=-1
00700       DO 20I=1,N
00800       II=II+1
00900       MM=N-II
01000       DO 20J=1,MM
01100       IJ=(I-1)*N+J
01200       B(IJ)=.93
01300 20    CONTINUE
01400       DO 30I=1,N
01500       DO 30J=1,I
01600       IJ=(I-1)*N+J
01700       B(IJ)=.93
01800 30    CONTINUE
01900       NBY2=N/2
02000       DO 40I=1,NBY2
02100       DO 40J=1,I
02200       IJ=(I-1)*N+J
02300       B(IJ)=.45
02400 40    CONTINUE
02500       IND=-1
02600       DO 50I=17,N
02700       IND=IND+1
02800       MM2=NBY2-IND
02900       DO 50J=1,MM2
03000       IJ=(I-1)*N+J
03100       B(IJ)=.45
03200 50    CONTINUE
03300       DO 60I=15,18
03400       DO 60J=15,18
03500       IJ=(I-1)*N+J
03600       IF(B(IJ).EQ..93)B(IJ)=0.0
03700       IF(B(IJ).EQ.0.45)B(IJ)=.93
03800 60    CONTINUE
03900       CALL PRF(B,D)
04000       STOP
04100       END
```

```
00100    C     ------------------------------------------------------------
-----
00200    C     ------------------------------------------------------------
-----
00300    C
00400    C     GENERATION AND RECONSTRUCTION OF A 3-DENSITY FIGURE WITH ORTH.
00500    C
00600    C     ------------------------------------------------------------
------
00700    C     ------------------------------------------------------------
------
00800          DIMENSION B(1024)
00900          N=32
01000          DO 10I=1,1024
01100    10    B(I)=0.45
01200          II=-1
01300          DO 20I=1,N
01400          II=II+1
01500          MM=N-II
01600          DO 20J=1,MM
01700          IJ=(I-1)*N+J
01800          B(IJ)=.93
01900    20    CONTINUE
02000          DO 30I=1,N
02100          DO 30J=1,I
02200          IJ=(I-1)*N+J
02300          B(IJ)=.93
02400    30    CONTINUE
02500          NBY2=N/2
02600          DO 40I=1,NBY2
02700          DO 40J=1,I
02800          IJ=(I-1)*N+J
02900          B(IJ)=.45
03000    40    CONTINUE
03100          IND=-1
03200          DO 50I=17,N
03300          IND=IND+1
03400          MM2=NBY2-IND
03500          DO 50J=1,MM2
03600          IJ=(I-1)*N+J
03700          B(IJ)=.45
03800    50    CONTINUE
03900          DO 60I=15,18
04000          DO 60J=15,18
04100          IJ=(I-1)*N+J
04200          IF(B(IJ).EQ..93)B(IJ)=0.0
04300          IF(B(IJ).EQ.0.45)B(IJ)=.93
04400    60    CONTINUE
04500          CALL PRFO(B,N)
04600          STOP
04700          END
04800    C
04900    CC
05000    CC
05100    C
```

```
C     ********************************************************************
C     ********************************************************************
C     GENERATION & RECONSTRUCTION OF A GRADED DENSITY SQUARE.
C     ********************************************************************
C     ********************************************************************


C     B  IS THE MATRIX WHERE THE PICTURE IS GENERATED
C     D1,D2,D3,D4 ARE THE DENSITIES USED.
C
      DIMENSION B(1024)
      N=32
      NN=N*N
      D1=0.;D2=.56;D3=.67;D4=1.00
      CALL SQUARE(B,N,D1,D2,D3,D4)
      CALL PRF(B,N)
      STOP
      END
C     SUBROUTINE SQUARE GENERATES THE SQUARE.
      SUBROUTINE SQUARE(B,N,D1,D2,D3,D4)
      DIMENSION B(1024)
      NN=N*N
      DO 10J=1,32
      DO 10I=1,32
      IJ=(I-1)*N+J
      IF(J.GT.8)GO TO 20
      B(IJ)=D1
      GO TO 10
   20 IF(J.GT.16)GO TO 30
      B(IJ)=D2
      GO TO 10
   30 IF(J.GT.24)GO TO 40
      B(IJ)=D3
      GO TO 10
   40 B(IJ)=D4
   10 CONTINUE
      RETURN
      END
C     ------------------------------------------------------------------
```

```
C      -------------------------------------------------------------------
C      SUBROUTINE PICOUT GIVES AN IMAGE OF THE ARRAY B ON COMPUTER
C      OUTPUT PAPER WHERE THE DISTINCT GREY LEVELS ARE ACCOMPLISHED
C      BY OVERPRINTING.
C      -------------------------------------------------------------------
C
C         B ---     MATRIX TO BE IMAGED.
       SUBROUTINE PICOUT(B,N,NN)
       DIMENSION B(1024)
       DIMENSION LN(130,7),DEN(21)
       DIMENSION RLEG(11)
       INTEGER G(21,7),NUM(21)
       DATA G(1,1),G(2,1),G(3,1),G(4,1),G(5,1),G(6,1),G(7,1),G(8,1),
      1G(9,1),G(10,1),G(11,1),G(12,1),G(13,1),G(14,1),G(15,1),G(16,1),
      2G(17,1),G(18,1),G(19,1),G(20,1),G(21,1)/1H ,1H-,1H=,1H+,1I,
      31HI,1HZ,1HX,1HA,1HM,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,
      41HO/
       DATA G(1,2),G(2,2),G(3,2),G(4,2),G(5,2),G(6,2),G(7,2),G(8,2),
      1G(9,2),G(10,2),G(11,2),G(12,2),G(13,2),G(14,2),G(15,2),G(16,2),
      2G(17,2),G(18,2),G(19,2),G(20,2),G(21,2)/1H ,1H ,1H ,1H ,1H ,
      31H ,1H ,1H ,1H ,1H ,1H-,1H=,1H+,1H+,1H+,1H+,1HX,1HX,1HX,1H ,
      41HX/
       DATA G(1,3),G(2,3),G(3,3),G(4,3),G(5,3),G(6,3),G(7,3),G(8,3),
      1G(9,3),G(10,3),G(11,3),G(12,3),G(13,3),G(14,3),G(15,3),G(16,3),
      2G(17,3),G(18,3),G(19,3),G(20,3),G(21,3)/1H ,1H ,1H ,1H ,1H ,
      31H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H.,1H=,1H.,1H.,1H.,1H.,1H.,
      41H./
       DATA G(1,4),G(2,4),G(3,4),G(4,4),G(5,4),G(6,4),G(7,4),G(8,4),
      1G(9,4),G(10,4),G(11,4),G(12,4),G(13,4),G(14,4),G(15,4),G(16,4),
      2G(17,4),G(18,4),G(19,4),G(20,4),G(21,4)/1H ,1H ,1H ,1H ,1H ,
      31H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H=,1H-,1HH,1HH,1HH,
      41HH/
       DATA G(1,5),G(2,5),G(3,5),G(4,5),G(5,5),G(6,5),G(7,5),G(8,5),
      1G(9,5),G(10,5),G(11,5),G(12,5),G(13,5),G(14,5),G(15,5),G(16,5),
      2G(17,5),G(18,5),G(19,5),G(20,5),G(21,5)/1H ,1H ,1H ,1H ,1H ,
      31H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1HC,1HB,1HH,1H-
      4/
       DATA G(1,6),G(2,6),G(3,6),G(4,6),G(5,6),G(6,6),G(7,6),G(8,6),
      1G(9,6),G(10,6),G(11,6),G(12,6),G(13,6),G(14,6),G(15,6),G(16,6),
      2G(17,6),G(18,6),G(19,6),G(20,6),G(21,6)/1H ,1H ,1H ,1H ,1H ,
      31H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
      41HV,1HV/
       DATA G(1,7),G(2,7),G(3,7),G(4,7),G(5,7),G(6,7),G(7,7),G(8,7),
      1G(9,7),G(10,7),G(11,7),G(12,7),G(13,7),G(14,7),G(15,7),G(16,7),
      2G(17,7),G(18,7),G(19,7),G(20,7),G(21,7)/1H ,1H ,1H ,1H ,1H ,
      31H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
      41HA/
       DATA DEN/.0,.15,.22,.25,.29,.33,.37,.40,.42,.45,.53,.56,0.6,
      1         0.4,0.7,.79,.85,.89,.93,.97,1.0/
       DATA NUM/1,1,1,1,1,1,1,1,1,1,2,2,2,3,3,4,4,5,5,6,7/
       DATA CHAR/0.6/
       DATA IBLANK,ISTAR/1H ,1H*/
       LINE=0
       IF(LINE.NE.0)PRINT 60
       IF(LINE.EQ.0)PRINT 61
       MLINE=130
       IF(LINE.EQ.0)MLINE=60
       XMIN=B(1);XMAX=B(1)
       XSUM=0.
       DO 16I=1,N
       DO 16J=1,N
       IJ=(I-1)*N+J
       XSUM=XSUM+B(IJ)
       IF(XMIN-B(IJ))12,12,10
10     XMIN=B(IJ)
12     IF(XMAX-B(IJ))14,16,16
14     XMAX=B(IJ)
16     CONTINUE
       N1=FLOAT(N)*CHAR
       XDIST=FLOAT(N)/FLOAT(N1+1)
       IF((XMAX-XMIN).GT.1.E-10)GO TO 18
       IF(LINE.NE.0)PRINT 72
       RETURN
18     IF((N+2).GT.MLINE)RETURN
       FRAC=1./(XMAX-XMIN)
```

```
      IF(M.GT.100)GO TO 22
      NOG=(60-.1)/4-1
      IF(LINE.EQ.0)NOG=2
      DO 20I=1,NOG
      PRINT 68
20    CONTINUE
22    DO 241=1,130
      DO 24J=1,7
      LN(I,J)=IBLANK
24    CONTINUE
      IMIN=(NLINE-J)/2
      IMAX=IMIN+M-1
      II1=IMIN-1
      II2=IMAX+1
      DO 261=III,II2
      LN(I,1)=ISTAR
26    CONTINUE
      PRINT 74,(LN(I,1),I=1,NLINE)
      PRINT 86
      Y=FLOAT(N)+.5
      DO 46K=1,M1
      Y=Y-XDIST
      J1=Y
      Y1=FLOAT(J1)
      J2=J1+1
      I1=0
      LMAX=0
      DO 40I=IMIN,IMAX
      I1=I1+1
      I1J1=(I1-1)*N+J1
      I1J2=(I1-1)*N+J2
      D=(B(I1J1)+B(I1J2)-B(I1J1))*(Y-FLOAT(J1))-XMIN)*FRAC
      DO 28M=1,21
      IF(D-DEN(M))30,36,28
28    CONTINUE
30    M1=M-1
      M2=M
      T=(DEN(M2)+DEN(M1))/2.
      IF(D-T)32,32,34
32    L=M1
      GO TO 38
34    L=M2
      GO TO 38
36    L=M
38    DO 40J=1,7
      IF(L.GT.LMAX)LMAX=L
      LN(I,J)=G(L,J)
40    CONTINUE
      NJ=NLN(LMAX)-1
      IF(NJ.LE.0) GO TO 44
      DO 42J=1,NJ
      PRINT 76,(LN(I,J),I=1,NLINE)
42    CONTINUE
44    NJ1=NJ+1
      PRINT 76,(LN(I,NJ1),I=1,NLINE)
46    CONTINUE
      DO 48I=II1,II2
      LN(I,1)=ISTAR
48    CONTINUE
      PRINT 74,(LN(I,1),I=1,NLINE)
      MLEG=MLINE/11
      NLEG=22/MLEG
      DO 561I=1,NLEG
      DO 50J=1,7
      DO 50I=1,130
      LN(I,J)=IBLANK
50    CONTINUE
      DO 52J=1,11
      ISUB1=(J-1)*11+1
      ISUB2=(J-1)+(I1-1)*NLEG
      ISUB3=ISUB2+1
      IF(ISUB2.LE.0)ISUB2=1
      IF(ISUB3.GT.21)ISUB3=21
      RLEG(J)=(DEN(ISUB2)+DEN(ISUB3))/(FRAC*2.)+XMIN
      DO 52K=1,7
```

```
      LN(ISUB1,K)=G(ISUB2,K)
52    CONTINUE
      DO 54I=1,6
54    PRINT 76,(LN(J,I),J=1,LLEN)
      PRINT 78,(G (I,7),J=1,LLEN)
      PRINT 80,(RLEG(J),J=1,LLEG)
56    CONTINUE
      IF(LINE.NE.0)GO TO 64
      DO 58J=1,7
      DO 581=1,130
      LN(I,J)=BLANK
58    CONTINUE
      RLEG(1)=DEN(21)/FRAC+XMIN
      DO 60K=1,7
      LN(1,K)=G(21,K)
60    CONTINUE
      PRINT 68
      DO 621=1,6
62    PRINT 76,LN(1,I)
      PRINT 78,LN(1,7)
      PRINT 80,RLEG(1)
64    CONTINUE
      IF(LINE.NE.0)PRINT 72
      RETURN
66    FORMAT(6H1*DVF*)
68    FORMAT(/)
72    FORMAT(6H1*DVN*)
74    FORMAT(1H ,30X,100A1)
76    FORMAT(1H+,30X,100A1)
78    FORMAT(1H ,30X,100A1)
80    FORMAT(11(1X,E10.4))
86    FORMAT(/)
      END
```

```
C
C     SUBROUTINE FOR OPRA
C     -----------------------------------------------------------------
C
C
C     REFERENCES:
C                       A        THE VECTOR IN WHICH THE WEIGHTS ARE STORED.
C                       C        THE PICTURE MATRIX.
C                       F        THE RECONSTRUCTED PICTURE MATRIX.
C                       N        N*N IS THE SIZE OF THE GRID.
C                       L        THE VECTOR WHICH KEEPS TRACK OF THE PRO-
C                                JECTION ELEMENTS.
C                       THETA ANGLE AT WHICH THE PROJECTION IS STARTED.
C
C
C     SUBROUTINES CALLED:
C             PROJT : SUBROUTINE WHICH ... DESIRED ANGLE.
C             PROD : OBTAINS INNER PRODUCT OF ...
C             ORTH : PERFORMS ORTHOGONALIZATION.
C             PICOUT : PRINTS OUT THE IMAGE.
C             ERR : CALCULATES ERROR MEASURES.
C
      SUBROUTINE PROJ(C, )
      DIMENSION A(1024),AA(1024),K(1024),(00),K (00)
      REAL C(1024),B,BB,F(1024)
      INTEGER P,PP,S
      REAL L(20)
      PRINT 1
1     FORMAT(1H ,'RECONSTRUCTION WITH ORTHOGONALISATION.'//)
      PI=3.1428570
      NPR=9
      DO 5LL=1,4
      MARK=LL
      TYPE*,MARK
      IF(LL.LT.2)GO TO 10
      L(1)=0.;L(2)=0.;L(3)=0.;L(4)=0.;L(5)=0.;L(6)=0.;L(7)=0.;L(8)=0.
      L(9)=0.;L(10)=0.;L(11)=0.;S=0
10    THETA=0.;S=1
      IF(L(S).LT.0.5)GO TO 20
      IF(L(S).GT.32.)GO TO
      GO TO 70
20    L(S)=L(S)+.5
      CALL PROJT(THETA,C,L,N,A,B,K,P,S)
      IF(P.LE.1)GO TO 20
      IF(MARK.EQ.1)   =   +1
      DO 40I=1,1024
40    AA(I)=A(I)
      BB=B
      PP=P-1
      DO 60I=1,PP
60    KK(I)=K(I)
70    THETA=THETA+PI/9.;S=S+1
      IF(S.GT.9)GO TO 160
      IF(L(S).GE.32)GO TO 70
80    L(S)=L(S)+.5
      DO 90I=1,1024
90    A(I)=0.
95    CONTINUE
      CALL PROJT(THETA,C,L,N,A,B,K,P,S)
      IF(L(S).GT.32)GO TO 70
      IF(P.LE.1)GO TO 80
      IF(MARK.EQ.1)   =   +1
      CALL ORTH(AA,  ,  ,  )
      CALL PROD(AA,F,Y)
      CALL PROD(AA,AA,Z)
      IF(Z.EQ.0.)GO TO 110
      DO 100I=1,PP
      F(KK(I))=F(KK(I))-(Y-BB)/Z*AA(KK(I))
100   IF(F(KK(I)).LT.0.)F(KK(I))=0.
110   DO 120I=1,1024
120   AA(I)=A(I)
      BB=B
      PP=P-1
```

```
      DO 140I=1,PP
140   KK(I)=K(I)
160   LSS=0
      DO 170S=1,9
170   IF(L(S).GE.32)LSS=LSS+1
      IF(LSS.EQ.9)GO TO 180
      IF(S.LT.9)GO TO 70
      GO TO 10
180   CONTINUE
200   FORMAT(1H ,32F5.2)
      THETA=0.;S=1
      L(S)=0.
220   L(S)=L(S)+.5
      CALL PROJT(THETA,C,D,N,A,B,K,P,S)
      IF(P.LE.1)GO TO 220
      CALL ORTH(AA,BB,A,B)
      CALL PROD(AA,F,Y)
      CALL PROD(AA,AA,Z)
      DO 240I=1,PP
      F(KK(I))=F(KK(1))-(Y-bb)/Z*AA(KK(I))
240   IF(F(KK(I)).LT.0.)F(KK(I))=0.
      PRINT 300,MARK,NPR,NEQ
300   FORMAT(14 ,'FOR ',I4,' ITERATION(S) ',I4,'PROJECTIONS ',I4,
     1'EQUATIONS '/)
      CALL PICOUT(F,C,1024)
      MIS=0
      DO 61=1,1024
      MF(I)=0
      IF(F(I).LT..30)MF(I)=0.
      IF(F(I).GE..30.AND.F(I).LE..68)MF(I)=.45
6     IF(F(I).GT..68)MF(I)=.93
      DO 71=1,1024
7     IF(C(I).NE.MF(I))MIS=MIS+1
      PRINT 222,MIS
      TYPE 222,MIS
222   FORMAT(1H ,'NO. OF MISMATCHES = ',I5)
      CALL ERR(C,F,N)
      PRINT 244
244   FORMAT(1H ,///)
5     CONTINUE
      RETURN
      END
```

```
C-------------------------------------
      SUBROUTINE PROD(PP,RR,DD)
C-------------------------------------
      DIMENSION PP(1024),RR(1024)
      DD=0.
      DO 10I=1,1024
10    DD=DD+PP(I)*RR(I)
      RETURN
      END
C-------------------------------------
C:    FOR ORTHOGONALIZATION
C-------------------------------------
      SUBROUTINE ORTH(AA,BB,A,B)
      DIMENSION AA(1024),A(1024)
      REAL B,BB
      CALL PROD(AA,A,YZ)
      CALL PROD(A,A,ZY)
      DO 10I=1,1024
10    AA(I)=AA(I)-YZ/ZY*A(I)
      BB=BB-YZ/ZY*B
      RETURN
      END
```

```
C-----------------------------------------------------------------------
      SUBROUTINE PRA(C,*)
C-----------------------------------------------------------------------
      DIMENSION A(1024),F(1024),K(600),C(1024),MF(1024)
      INTEGER P,PP,S
      READ L(20),A
      MPR=6
      I=32
      PI=3.1428570
      DO 5LL=1,11
      MARK=LL
      TYPE*,MARK
      IF(LL.LT.2)GO TO 10
      IF(LL.LT.2)GO TO 10
      L(1)=0;L(2)=0;L(3)=0;L(4)=0;L(5)=0;L(6)=0;L(7)=0;L(8)=0
      L(9)=0;L(10)=0;S=0
10    THETA=0.;S=1
      IF(L(S).LT..5)GO TO 20
      IF(L(S).LT.32)GO TO 80
      GO TO 70
20    L(S)=L(S)+.5
      CALL PROJ(THETA,C,L,N,A,P,K,P,S)
      IF(P.LE.1)GO TO 20
      IF(MARK.EQ.1)NEQ=NEQ+1
      PP=P-1
      CALL PROD(A,F,Y)
      CALL PPOD(A,A,Z)
      IF(Z.EQ.0.)GO TO 7A
      DO 100I=1,PP
      F(K(I))=F(K(I))-(Y-B)/Z*A(K(I))
      IF(F(K(I)).LT.0.)F(K(I))=0
      THETA=THETA+PI/6.;S=S+1
      IF(S.GT.6)GO TO 100
      IF(L(6).GE.32)GO TO 70
      L(S)=L(S)+.5
      DO 90I=1,1024
      F(I)=0.
      CALL PROJ(THETA,C,L,N,A,B,K,P,S)
      IF(L(S).GT.32)GO TO 70
      IF(P.LE.1)GO TO 80
      IF(MARK.EQ.1)NEQ=NEQ+1
      PP=P-1
      CALL PROD(A,F,Y)
      CALL PROD(A,A,Z)
      IF(Z.EQ.0.)GO TO 100
      DO 110I=1,PP
      F(K(I))=F(K(I))-(Y-B)/Z*A(K(I))
100   IF(F(K(I)).LT.0.)F(K(I))=0.
      NSS=0
      DO 17 S=1,6
17    IF(L(S).GE.32)NSS=NSS+1
      IF(NSS.EQ.6)GO TO 180
      IF(S.LT.6)GO TO 70
      GO TO 100
180   PRINT 300,N,P,NEQ,MARK
300   FORMAT(1H ,' NPR = ',I4,'  NEQ = ',I6,'  ITERATION(S) = ',I4/)
      PRINT 200,(F(I),I=1,1024)
200   FORMAT(//)
C     CALL PTCOM(F,J,1024)
      PRINT 90
      NIS=0
      DO 6I=1,1024
      IF(F(I).GE..5)F(I)=1
      DO 7I=1,1024
7     IF(A(I).NE.F(I))NIS=NIS+1
      PRINT 240,NIS
240   FORMAT(1H ,'NO. OF MISMATCHES = ',I5)
      PRINT 400
      CALL ERR(C,P,N)
5     CONTINUE
      RETURN
      END
```

```
001 )     C-------------------------------
002               SUBROUTINE ERR(C,F,N)
003       C-------------------------------
004               DIMENSION C(1024),F(1024),FF(1024)
005               REAL MRERR
006               XN=N*N
007               SUM0=0.;SUM1=0.;SUM2=0.;SUM3=0.
008               DO 10I=1,N
009               SUM0=SUM0+C(I)
010               SUM1=SUM1+(C(I)-F(I))**2
011               SUM2=SUM2+ABS(C(I)-F(I))
012       10      CONTINUE
013               AVG=SUM0/XN
014               DO 20I=1,N
015               SUM4=SUM4+(C(I)-AVG)**2
016       20      SUM3=SUM3+(F(I)-AVG)**2
017               DELTA=SQRT(SUM1/XN)
018               MRERR=SUM2/SUM0
019               VAR=SUM3/XN
020               RMS=SQRT(SUM1/SUM4)
021               PRINT 300
022               PRINT 400
023               PRINT 500,DELTA,MRERR,VAR,RMS
024       300     FORMAT(///)
025       400     FORMAT(1H ,' THE ERROR MEASURES ARE : ',/)
026       500     FORMAT(1H ,'DELTA = ',F8.5,'   MRERR = ',F8.5,'   VAR = ',
027               1F8.5,'   RMS = ',F8.5)
028               DO 35I=1,1024
029               IF(F(I).GE..5)FF(I)=1.0
030       35      IF(F(I).LT..5)FF(I)=0.
031               SUMM0=0.;SUMM1=0.;SUMM2=0.;SUMM3=0.;SUMM4=0.
032               DO 15I=1,N
033               SUMM0=SUMM0+C(I)
034               SUMM1=SUMM1+(C(I)-FF(I))**2
035               SUMM2=SUMM2+ABS(C(I)-FF(I))
036               CONTINUE
037               AVG=SUMM0/XN
038               DO 25I=1,N
039               SUMM4=SUMM4+(C(I)-AVG)**2
041       25      SUMM3=SUMM3+(FF(I)-AVG)**2
041               AMDELTA=SQRT(SUMM1/XN)
042               AMRERR=SUMM2/SUM0
043                 AR=SUMM3/XN
044                 S=SQRT(SUMM1/SUM4)
045                 T 35
045                 (I 45)
047                 T= 55,AMDELTA,AMRERR,AVAR,ARMS
048       3         (///)
049       4         IC I ,' THE ERROR MEASURES ARE : ',/)
050       55        (1H ,'AMDELTA = ',F8.5,'   AMRERR = ',F8.5,'   AVAR = ',
051               1F8.5,'   RMS = ',F8.5)
052               RETURN
053               END
```